User's Guide™

THE MAGAZINE FOR CP/M & MS-DOS COMPUTER USERS

No. 10

$4

Computer Languages

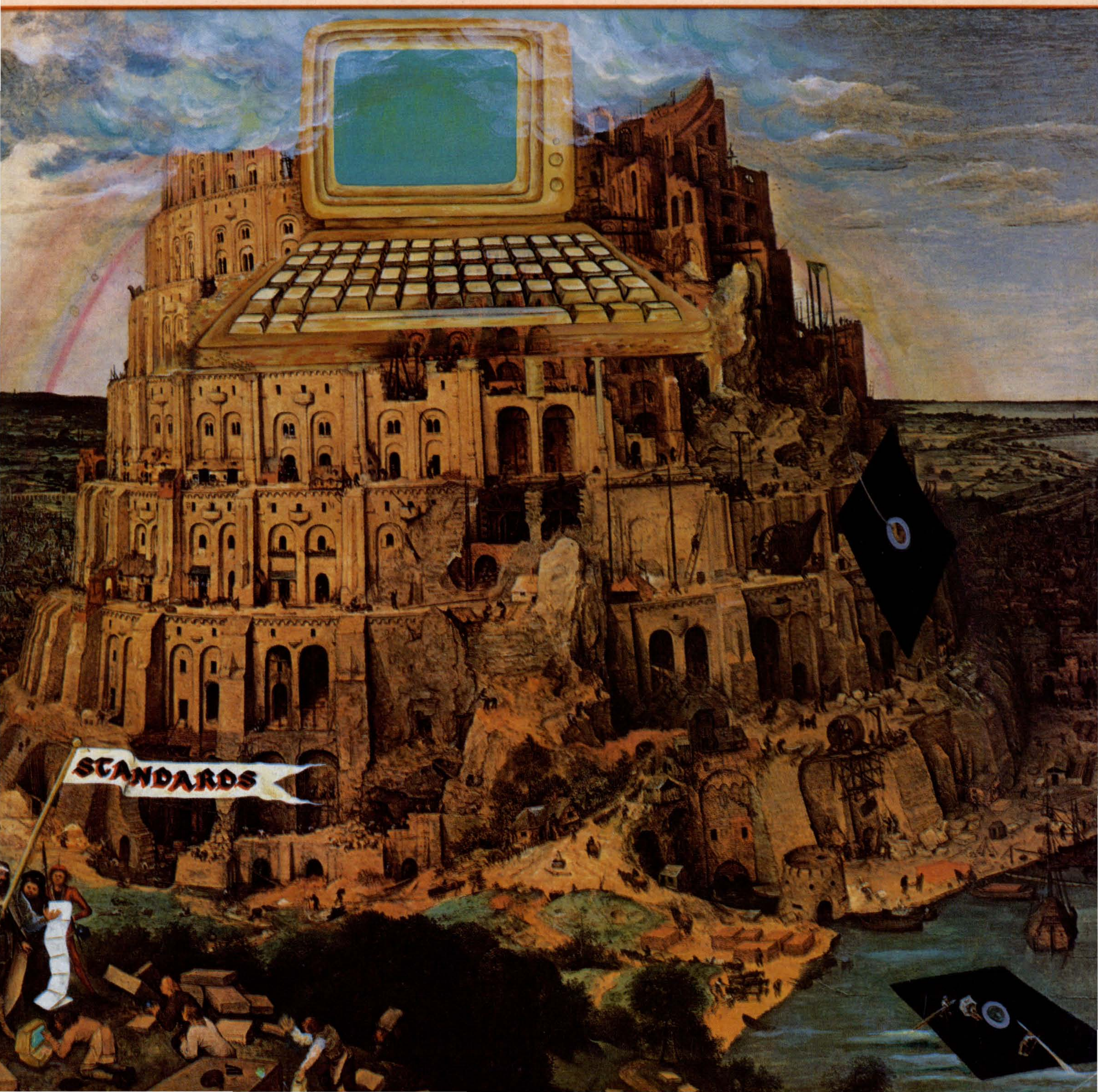
MBASIC and CBASIC

Turbo Pascal

CB-80 & CB-86

Batch Processing

Free Utilities



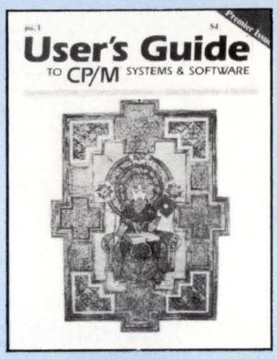
Back Issues

Get them now
before they sell out!

> "So far I haven't found one [issue] that wasn't worth more than what it costs. Recommended."
> — Jerry Pournelle, *Byte* (Dec. 1983).

Issue #1R: CP/M® (revised issue)

Learning to Use CP/M • Impatient User's Guide® to CP/M • Using CompuServe® • Preventing Computer Disasters •


no. 1
$4
Premier Issue
User's Guide
TO CP/M SYSTEMS & SOFTWARE

Issue #2: WordStar® & SuperCalc®

A Guide to Using WordStar • Impatient User's Guide to WordStar • WordStar Installation Guide • A Guide to Using SuperCalc • Disk Doctor: The Rx For Bad Disks • For Starters: Word Processing & Spreadsheets •

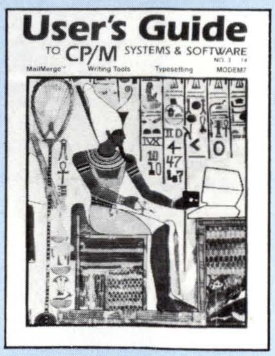
Issue #3: MailMerge & Modem7

MailMerge: WordStar's Printing Robot • Impatient User's Guide to MailMerge • Using DocuMate Plus and The Word Plus • Using Footnote and Bibliography • Modem7 Tutorial and RCPM List • WordStar File and Print Summary • For Starters: Word Processing Power •

User's Guide
TO CP/M SYSTEMS & SOFTWARE
•WordStar
•Disk Doctor
•SuperCalc

Issue #4: CP/M Plus

Overview of CP/M Plus • Guide to SuperCalc (Pt. 2) • SuperCalc SuperSummary • CP/M Plus Operating Tools • Reference Guide to DU (Disk Utility) • Overview of Floppy Disks • System Overview: CompuPro 8:16 • For Starters: Exploring dBASE II •


User's Guide
TO CP/M SYSTEMS & SOFTWARE
MailMerge
Writing Tools
Typesetting
MODEM7

Please send me the following back issues:

Payment by charge card: ☐ Visa ☐ Mastercard   Payment must accompany order.

Signature required: ______________ Exp. date ________ MC ________

Card # ______________________________

☐CP/M Summary Card, $2 each: # of cards ______ = $______________

☐Back issue #1 at $5 each, issue #2 at $6 each, and all other issues at $4 each:

Issue# 1 Qty. __________ at $5 each ($6.50 1stC or foreign, $11 air) = $__________

Issue# 2 Qty. __________ at $6 each ($7.50 1stC or foreign, $12 air) = $__________

Issue# ___ Qty. __________ at $4 each ($5.50 1stC or foreign, $10 air) = $__________

TOTAL ORDER: $__________

Name ______________________________

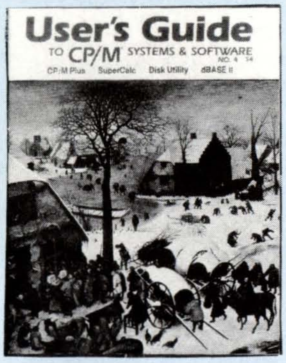
Company ______________ Title ______________

Address ______________________________

City, State, Zip ______________________________

☐ Check if you do not want promotional mailings.

Make checks payable to: User's Guide, Offer expires 12/31/85
Subscription Dept.: P.O. Box 5245, Redwood City, CA 94063.
Please allow 6-8 weeks for processing.


User's Guide
TO CP/M SYSTEMS & SOFTWARE
NO. 4 $4
CP/M Plus SuperCalc Disk Utility dBASE II

Issue #5: dBASE II

Introduction to dBASE II • dBASE II Application: Subscription Fulfillment • Perfect Filer Overview & Starter • Using CP/M Plus (Pt. 1) • CP/M Plus: Inner Workings • Hard Disks For Small Computers • WordStar on the TeleVideo 950 • System Overviews: Radio Shack Model 100, PMC MicroMate •

Issue #6: Communications

CompuServe Electronic Mail • CompuServe Public Access & File Transfers • Communications Programs: COMMX and MITE • Guide to Using WordStar (Pt. 2) • Converting From Perfect Writer to WordStar • WordStar vs. Perfect Writer • CP+ and Organizer: Menus for CP/M • Cache/Q: A System Enhancer • Glossary of Computer Communications Terms • For Starters: Communications •

User's Guide
TO SYSTEMS & SOFTWARE
NO. 5 $4
CP/M Plus Tutorial dBASE II Tutorial Perfect Filer
Radio Shack Model 100 PMC MicroMate TeleVideo & WordStar Hard Disks

Issue #7: T/Maker III & CP/M Plus

T/Maker III: Integrated Text Processor, Spreadsheet and Table Maker • Using CP/M Plus, Pt. 2: Utilities For Automation • FILEBASE for Personal Computers • Proportional Printing With Perfect Writer • MS-DOS and PC-DOS Overview, Pt. 1 • Making Communications Connections • For Starters: Brief Tour of CP/M Systems • System Overview: DEC Rainbow • **Annual Index for issues 1-6** •

Issue #8: Accounting Packages Compared

Show Report: Softcon and 9th West Coast Computer Faire • For Starters: Computerized Accounting • Small Business Bookkeeping: Taranto's General Ledger • MS-DOS and PC-DOS, Part 2 • Comparing Accounting Packages: Accounting Plus vs. SBT • Using Mini-Ledger • Patching With DDT and SAVE • dBASE II Tutorial, Part 2: Indexing and Sorting • ASCII Chart •

Issue #9: Multi-User and Concurrent Systems

For Starters: Multiple Users and Multiple Computers • The Sleaze Patrol • The Meaning of Concurrency • Concurrent CP/M-86 and MP/M-86: Files, Disks, Spooling and Printing • Concurrent CP/M-86: Versions for DEC Rainbow, IBM PC • MP/M-86: Version for CompuPro • NewWord Evaluation • Bible Study by Computer • Punctuation & Style Evaluation • System Overviews: OSM Zeus 4 Network, Ampro Bookshelf Computer •

Act now to complete your set of *User's Guide* back issues! Limited offer while supplies last! This may be your last chance to get a complete set!

Limited Offer:
Issues #1-#6 for $20
Issues #1-#10 for $30

☐ SPECIAL OFFER: Issues 1-6 For $20 $________
☐ SPECIAL OFFER: Issues 1-10 For $30 $________
Payment must accompany order. TOTAL ENCLOSED: $________
We accept Visa and Mastercard. Phone: 415-364-0108.
Name ____________________
Company ____________ Title ____________
Address ____________________
City, State, Zip ____________________
☐ Check if you are already a subscriber, or are sending in a subscription card.

Make checks payable to: User's Guide. Offer expires 2/1/85.
Back Issues Dept.: P.O. Box 5245, Redwood City, CA 94063.
Please allow 6-8 weeks for processing.

User's Guide is a trademark of TUG Inc.
CP/M is a registered trademark of Digital Research, Inc.
MS-DOS is a trademark of Microsoft.

CP/M is a registered trademark of DRI (Digital Research, Inc., Pacific Grove, CA). Concurrent CP/M is a trademark of DRI. CompuServe is a trademark of H & R Block. IBM is a registered trademark of International Business Machines. WordStar and MailMerge are trademarks of MicroPro Int. (San Rafael, CA). SuperCalc is a trademark of Sorcim Corp. (San Jose, CA). DocuMate Plus is a trademark of Orthocode Corp. (Albany, CA). dBASE II is a trademark of Ashton-Tate (Culver City, CA). Perfect Filer is a trademark of Perfect Software. Radio Shack is a trademark of Tany Corp. MicroMate is a trademark of PMC (Recortec, Sunnyvale CA).
User's Guide and Impatient User's Guide are trademarks of TUG, Inc.

User's Guide™

User's Guide Number 10 (Volume 2.4)
Sept/Oct 1984

User's Guide , ISSN 0747-8534, is published six times a year by TUG, Inc., Box 5245, Redwood City, CA 94063. Second-class postage paid at Redwood City, CA 94063 (USPS 689-550).

Entire contents Copyright © 1984 by TUG, Inc. All rights reserved. Reproduction of material appearing in *User's Guide* is forbidden without explicit permission. Send all reprint requests to Reprints, User's Guide, Box 5245, Redwood City, CA 94063.

SUBSCRIPTION INFORMATION:
Single copy price $4.00 ($5.50 foreign or U.S. first-class, $10 foreign airmail). Subscription for six issues (one year) is:

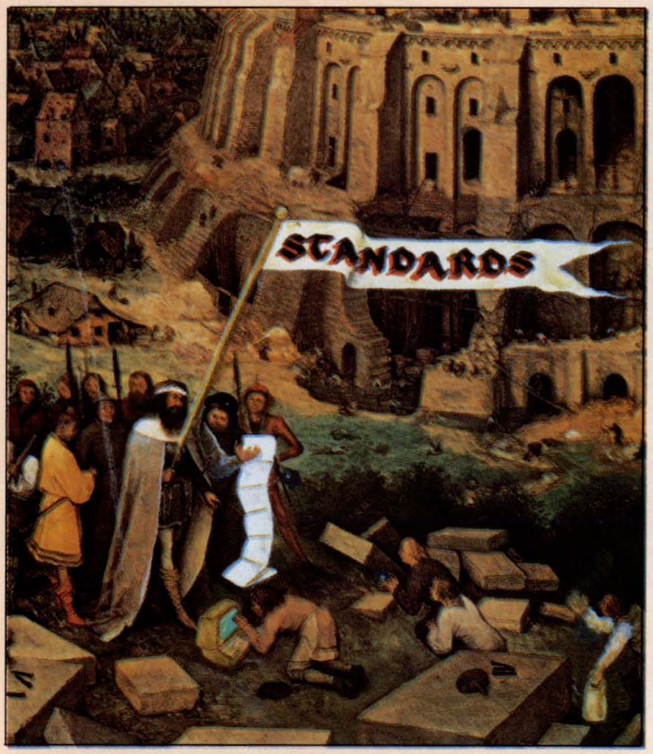
U.S. — $18 if payment accompanies order, $21 if subscription is billed (second class). Add $10 for first-class delivery.

CANADA & MEXICO — $22 in U.S. currency, payment must accompany order (second class). $32 for first-class delivery.

FOREIGN — $40 for air mail subscription. Payment (U.S. currency) must accompany foreign orders.

Inquiries about bulk, retail sale, and group rates are encouraged. Address inquiries to Circulation Dept., *User's Guide*, Box 5245, Redwood City, CA 94063. Allow 6-8 weeks for processing of order.

POSTMASTER: Please send address changes (Form 3579) to User's Guide, Box 5245, Redwood City, CA 94063.



CP/M and CP/M-86 are registered trademarks of Digital Research, Inc. IBM is a registered trademark of International Business Machines. MS-DOS is a trademark of Microsoft. WordStar and MailMerge are trademarks of MicroPro, Intl.
User's Guide and Impatient User's Guide are trademarks of TUG, Inc.
User's Guide (TUG, Inc.) is not affiliated with Digital Research, Microsoft, or any hardware or software manufacturer or distributor.

Contents...

EDITORIAL
Amateur Programming **4**

LETTERS **62**

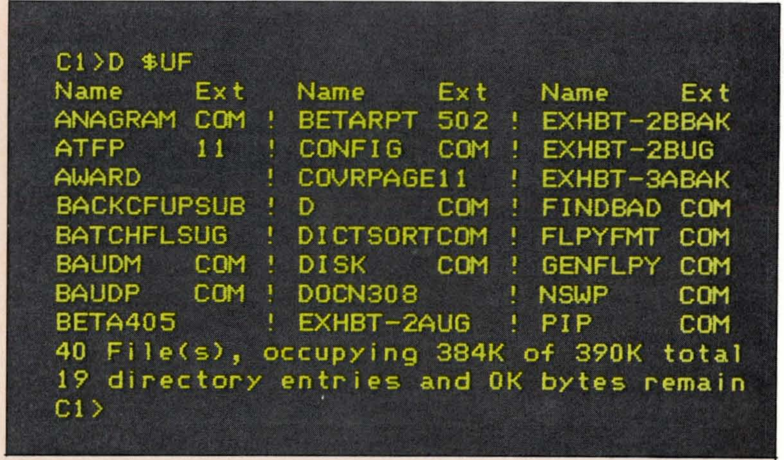
FOR STARTERS
Computer Languages **6**

Programming languages available for personal computers range from inexpensive packages for beginners to sophisticated ones for serious programmers.

PUBLIC DOMAIN

File Handling Utilities **19**

Making life easier with DISK.COM, NSWP.COM and D.COM, all from the CP/M and MS-DOS public domain libraries.
by Bob Wolff



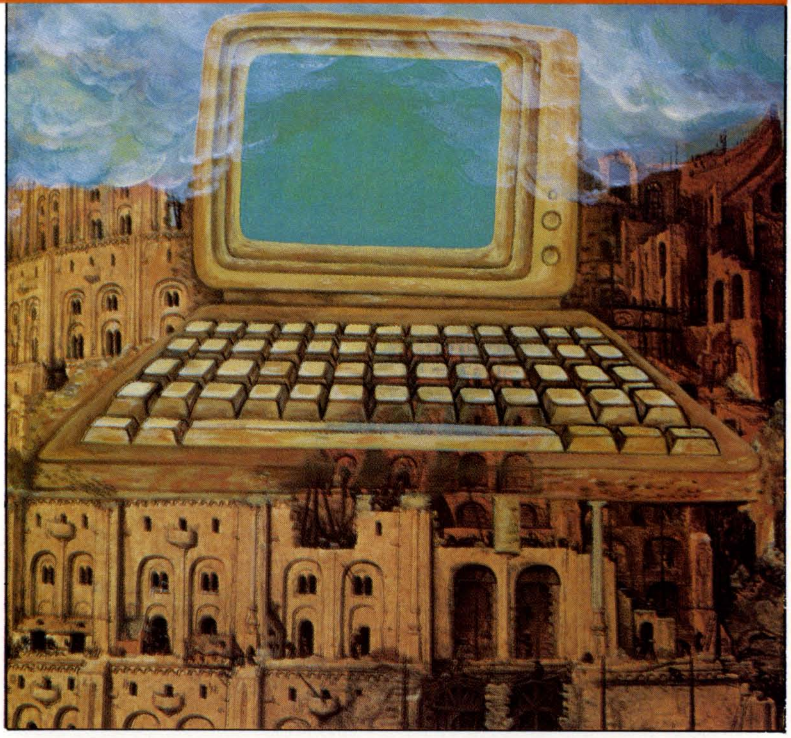
```
C1>D $UF
Name    Ext   Name    Ext   Name    Ext
ANAGRAM COM ! BETARPT 502 ! EXHBT-2BBAK
ATFP    11  ! CONFIG  COM ! EXHBT-2BUG
AWARD       ! COVRPAGE11  ! EXHBT-3ABAK
BACKCFUPSUB ! D       COM ! FINDBAD COM
BATCHFLSUG  ! DICTSORTCOM ! FLPYFMT COM
BAUDM   COM ! DISK    COM ! GENFLPY COM
BAUDP   COM ! DOCN308     ! NSWP    COM
BETA405     ! EXHBT-2AUG  ! PIP     COM
40 File(s), occupying 384K of 390K total
19 directory entries and 0K bytes remain
C1>
```

About the Cover

Brueghel's Tower of Babel, as visualized and painted by modern-day artist Sahnta Pannutti, is a not-so-subtle allegory for the computer industry's quest for the most knowledgeable and flexible thought machine (the ultimate tower). The computer industry has its roots in the research projects of the early fifties to make one large computer to serve all of the public and private sector needs.

What happened was confounding at first (a bizarre set of esoteric computer languages used by equally bizzare and eccentric programmers), but has now become an acceptable environment of productivity where many computer languages are not only understood but applied in efficient ways by people who are not "typical computer nerds."

Issue 10 (Vol. 2.4)



TUTORIALS

Running MBASIC Programs **8**
Learning to use the MBASIC interpreter gives you the opportunity to use thousands of free and relatively-inexpensive MBASIC programs.
by Michael Bradley

Turbo Pascal (Part One) **34**
The first installment presents a simple Pascal program and describes the Pascal compiler.
by Tony Bove & Cheryl Rhodes

From Source to Object: Executable Programs **44**
How to use ASM or ASM-86, with LOAD or GENCMD, to produce executable assembly language programs, even without training in assembly language.
by Tony Bove & Cheryl Rhodes

Batch Processing In CP/M and MS-DOS **52**
Creating your own mini-programs to handle multiple operations.
by Bob Wolff

ANNOUNCEMENTS

New Products **68**
Computers, add-on hardware, software, networks, peripherals, accessories, events...

ADVERTISER INDEX **76**

EDITORIAL Tony Bove, Cheryl Rhodes (chief editors)
Steve Rosenthal, Kelly Smith (contrib. editors)
Thomas Howe, Arthur Naiman,
Jonathan Sachs, Bob Wolff (writers)
Mary Crevelt (assistant)
ART Lon Shoemaker (cartoons),
Paul Winternitz (photos)
PRODUCTION Lon Shoemaker
CIRCULATION William Shoemaker, Kathy Slusser
ADVERTISING Annie Warren (coordinator)
(See advertiser index, pg. 76, for information.)

OVERVIEW

MS-DOS and PC-DOS (Part 3) **47**
An introduction to MS-DOS commands from a CP/M user's point of view.
by Steve Rosenthal

APPLICATION NOTE

Giving Data to a BASIC Program From the System Command Line **47**
If you can run the BASIC program from the CP/M and MS-DOS command line, you can also supply the program's data from that command line using this technique.
by David M. Alexander

EVALUATION

The CBASIC Compiler: CB-80 and CB-86 **39**
Digital Research's compilers for CBASIC are for serious programmers.
By Dale Blanchard

Amateur Programming

We have come out of the closet. We are "amateur" programmers, and we're not ashamed.

Fantasizing our IF/THEN/ELSE decisions and DO/WHILE routines, we are constantly stumbling over the syntax of the different computer languages but are still managing to produce working programs. Most of them do mundane things like printing mailing labels in postal zone order with subtotals, or converting data from one record format to another for use with a different system.

We hardly know enough to avoid infinite loops, we use GOTOs where we should re-design, we put in enough variables to make a general purpose program but fail to solve the specific problem, and we constantly fall victim to Murphy's laws. But we enjoy it. We're up late at night hacking away, convinced that we are solving our productivity problems by making our computers do more of the work. In most cases we are succeeding.

In the past we would never have admitted it. Real programmers might sneer at our un-mathematical approach to writing programs, but we were just like many of them — staying up late at night "hacking," ignoring television shows, sports events, political conventions, and our previous social lives.

Party-goers never missed us (assuming we'd gone to other more interesting parties). At parties we actually *did* go to, we'd find real programmers poking at Apple IIs or Macintoshs or IBM PCs, with super graphics, pull-down menus, and fast-moving mice, but hardly ever did we see someone demonstrating a CP/M machine.

CP/M? dBASE II? Mention them to serious programmers, the kind who talk about the graphics at the latest SIGGRAPH conference, if you want giggles and belly laughs.

So how could we possibly explain that we were writing dBASE II code? They wouldn't even call it code. To many of them, "hacking" is only done in languages like C or Forth.

We wrote programs for our own use, not for others. Our fetish was data management. We also liked converting print codes in text files into typesetting codes. BASIC was the language of necessity until recently, along with dBASE II for data management, WordStar for writing and editing, SuperCalc for producing spreadsheets, and T/Maker III for keeping numeric lists.

These programs didn't work well together, but they all have the capability to read and write simple ASCII text files. We kept some details in that common denominator, and converted data to more sophisticated formats whenever we had to (not all the time, like some of these newfangled "integrated" packages).

Our dBASE II "programs" might not be considered professional programs, but they work, and they were fun to build. Most important was the ability to access any field of any record in the dBASE II data base, directly, using dBASE II commands.

dBASE II is a good example of a data base management system that lets you write your own programs (a two-part tutorial on dBASE II appeared in previous issues, and we will continue to provide application notes). You can purchase a complete accounting system written in dBASE II, and if the system is provided in "source code" form, it means that the dBASE II programs were written in the command language and provided as text files you can edit. You can therefore modify the programs for any special requirements you have. dBASE II is a small command language that is easily understood in the context of managing data records, compared to "real" programming languages like BASIC or Pascal.

In your quest for the perfect application system, you should consider the package that provides enough functions to put you in control of your application.

□

This issue is dedicated to *amateur programmers*: problem solvers who did not receive formal training in programming and who are not employed as programmers. We are occasional hackers who tackle very specific jobs in order to improve our productivity. Usually the jobs involve moving data or making interesting correlations between two sets of data, and presenting the results on the screen and on paper.

Professional programmers may see nothing hard about programs that correlate data, print reports, print mailing labels or convert data to new formats. In fact, most programming texts present more complicated programs involving mathematical functions. In this issue we present simple programs that are easily understood and also useful.

The articles in this issue are useful for both professional and amateur programmers, but attention is paid more to the latter, making the articles accessible to a wider audience.

Editorial *continued on page 76.*

Your CP/M-80 becomes THE definitive 8-bit Operating System

ZCPR3 SETS YOU FREE!

- Command Line IF-ELSE-GOTO, Job Flow Control Automation
- File, Disk, Memory, and System Status Utilities, all online
- Named Directories, each with optional password access
- Dynamically loadable Resident Command Packages
- Error Handling Packages, graceful recoveries
- Screen-oriented Menu Generators
- Complete online HELP system
- SHELLS, with shell variables
- File Search Paths, organize your system
- Multiple Commands per Line, program chaining
- Aliases, eliminates complex keystroke entries, nestable
- Shell generator, make any application program a SHELL
- Terminal Install utility, uses UC Berkeley TERMCAP database

ZCPR3 permits highest productivity while providing flexible system control and maximum usability. It's terrific! The utilities create a tool-set environment that's hard to beat even by large main-frame machines. You are free to do what you want in the way you want. Try it and see if you don't agree. Newsletter, User Group and 24-hour bulletin board report ZCPR3 community activities.

1. CORE CP/M CCP REPLACEMENT MODULE Starter-kit utilities on 2 disks, plus 173-page SAMPLER documentation	only $39.00
2. ZCPR3 UTILITIES Complete, full source on 8 disks	$89.00
3. Z3-DOT-COM Auto-install ZCPR3. Load-and-go, complete full-up ready-to-run system on 4 disks	$149.00
4. ZCPR3: THE MANUAL Lavish, typeset, over 300 pages	$19.95
5. SYSLIB3 MACRO LIBRARY Used to write most ZCPR3 utilities, documention and full source on 4 disks	$29.00
6. DISCAT Menu-driven disk and file catalog system under ZCPR3	$49.00

VISA/MASTERCARD, check or money order accepted. Specify disk format desired. Add $3.00 shipping and handling. Calif. add 6.5% sales tax. Phone or send order now to Echelon, Inc. — your single source for ZCPR3 related software, support and documentation. We market programs taking full advantage of ZCPR3 capabilities; send us yours for evaluation and report.

Trademarks:
CP/M, Digital Research
ZCPR3, SYSLIB3, Richard L. Conn
Z3-Dot-Com, Alpha Systems Corp.
DISCAT, Echelon, Inc.

Echelon, Inc. 101 First Street • Los Altos, CA 94022 • 415/948-3820

Circle number 7 on the reader service card.

Computer Languages

Programming languages available for personal computers range from inexpensive packages for beginners to sophisticated ones for serious programmers.

Computer languages came into being along with our Twentieth Century Tower of Babel, the computer. The two are inseparable: the language tells the computer what to do. The dream of having one computer serve all of humanity was never to be realized; the computer was bulky, expensive, and programmed to do very specific tasks. The languages used were specific to each machine. Now it seems that we have a variety of computers, all in different sizes, all programmed in different languages.

Software is any set of computer instructions, and it may be the first human invention (after the spoken language itself) used to invent itself. Primitive software was used to invent more sophisticated software that could read statements made in a "programming language." This programming language was in turn used to build "operating systems." The operating systems were then used to create application programs and system programs.

Scientists found out very quickly that they needed software to make computers work, but it wasn't until computers were perceived to be useful in business that "standards" for creating software were developed and adhered to.

One of the early methods of communicating with a computer was through "toggle switches" on the front panel of the machine. These toggle switches let you load data and memory addresses using binary numbers — ones and zeroes in a pattern. A set of ones and zeros formed a *program.* One of the first personal computers, The Altair (one of the first personal computers, circa 1976) required the use of toggle switches to load the primitive instructions used to start up the computer.

When punch card machines were invented and began to be widely used, the toggle switches were given a rest. Punch cards made it possible to input more than ones and zeros, and codes were developed to represent letters of the alphabet. ASCII (American Standard Code for Information Interchange) is the one we use on personal computers, but there are others in use on larger computers. These codes were actually hexadecimal (base 16) or octal (base 8) numbers, not the usual numbers (base 10) you are used to counting with.

Terminals (keyboard/display units) soon replaced punch cards, and with them came more computer languages. With a keyboard you could easily type letters and numbers to represent instructions and data.

With terminals (keyboard/display units) came the possibility of standardizing at least the *method* of creating programs. The first real computer language, called *assembly language,* was invented for the class of computers existing at that time. As each new computer was invented, an assembly language of some sort was created to manipulate it.

The standard programming method still used today is to write a program as text on the screen, save the text in a disk file, and run a special program that changes the text into a program the computer can understand.

You use an *assembler* program to change an assembly language text file into a program. For other languages you use a translating program of some sort, such as an *interpreter* or a *compiler* (both explained below).

At first glance, assembly language looks harder to learn than Greek. Assembly language statements consist of acro-

by Tony Bove & Cheryl Rhodes

nyms and hexadecimal numbers. You type assembly language statements into an otherwise ordinary text file, with ordinary text as comments describing the instructions. The text file is then submitted to the assembler to be assembled into *object code* — instructions in a special hexadecimal code.

The object code is impossible to read. The hexadecimal numbers are actual computer instructions or ASCII codes representing the data. The object code resides in a disk file (with the extension ".HEX" or ".H86" in CP/M systems). You have to *load* the object code into the computer's random-access memory (RAM) in order to run the program.

Versions of assembly language are specific to that brand of CPU (central processing unit); for example, CP/M systems can run software written in Z80 assembly language or 8080/8085 assembly language, CP/M-86 systems run the 8086/8088 assembly language, and so on.

For a long while assembly language was the only language available for writing software such as the set of programs that make up an entire operating system. Assembly language is considered to be a powerful, bare-bones tool for writing programs that occupy a small amount of memory.

Assembly language programming takes a lot of time, so other languages were developed to speed up programming efforts. Many of the new languages consisted of words used to replace entire routines written in assembly language. The words made the writing of similar programs much easier. The new languages were dubbed "high-level" languages to distinguish them from assembly language.

FORTRAN (Formula Translator), one of the first high-level languages, was designed to allow scientists to express calculations in algebraic formulas. The language looks much like today's BASIC and is widely used on minicomputers and personal computers.

COBOL was designed by a committee attempting to standardize business computing on mainframe computers. COBOL has the distinct advantage over Fortran in its ability to define forms to emulate the paperwork of business and government.

Today COBOL is still used in business applications, but mostly in mainframe computers, because COBOL programs require more internal memory than is usually available in personal computers.

BASIC was developed to help teach students how to become programmers. It combines the formula expressions of FORTRAN with the print formatting of COBOL. Today there are many dialects of BASIC, which makes it difficult to transfer a BASIC program from one machine to another.

Pascal, a descendent of Algol, was designed to allow the programmer to write in a *structured* way. Routines in the program can be described in English as "if something is true, do this and then that, and if not true, do something else." Each "thing to do" can be written separately as a *procedure* and called from other routines. This method greatly simplified some programming efforts, thereby opening up possibilities for writing more elaborate software.

"Modular" programming, a technique utilized in Pascal and other structured languages, became the buzzword to describe the method of writing programs in modules that can be used in other programs, thus cutting down programming effort and time.

The C language combines the easy-to-read structure of Pascal with the power of assembly language, and is widely regarded as the language that is the easiest to use for a program to be transported from one machine to another. First developed for UNIX systems, C is also available for CP/M systems. Another C-like language, PL/I, is also structured and a powerful systems programming tool.

Other languages available for CP/M systems include Forth, Lisp, Ada, Modula 2 and Logo.

How It's Done

How do programmers write programs? In most cases the programmer uses a personal computer or large computer system to write the routines in a programming language such as C, Pascal or BASIC.

There are two kinds of personal computer programs: *interpretive* (sometimes called interactive) and *compiled*. The difference is in the way the language statements are translated.

Languages such as MBASIC (Microsoft) let you type each BASIC statement and see the result immediately so that you can test your program as you write it. MBASIC does this by using a translator that is *always active* and interpreting what you type as BASIC statements. MBASIC is called an *interpretive* BASIC.

On the other hand, languages such as CBASIC (DRI) and Pascal require that you write an entire program as a text file and then submit the text file to a translating program called a *compiler*. The statements are compiled into a compact program which can then be run. To test such programs, you use a tool called a *debugger* which lets you step through the program's statements as they execute.

Compiled programs generally run faster and are more compact than interpreted programs. The bulk of commercial business software available for CP/M systems was written in a compiled language (CBASIC is very popular for business applications).

To set up a system for serious programming, we recommend at least two disk drives, a good word processing program, and a dot matrix printer. You need at least 48K of internal memory in an eight-bit system, and at least 64K in a sixteen-bit system.

If you are designing programs using an eight-bit system, keep in mind that most programs must be converted into a sixteen-bit language in order to use them in a sixteen-bit system. Utility programs to help perform this conversion for CBASIC programs are described in the article on CBASIC, CB80 and CB86.

For help in designing serious programs for the CP/M user community, get in touch with Digital Research, Inc. DRI provides a support program of seminars and training for independent software vendors, and offers many programming languages and utilities for many different computers.

Running MBASIC Programs

You can use thousands of free and commercial MBASIC programs if you learn how to use The MBASIC Interpreter.

Nearly every personal computer is supplied with something called BASIC, which is included with the system software. BASIC is an acronym for Beginner's All-purpose Symbolic Instruction Code. It is a very popular *computing programming language*. Although you might not think of yourself as a computer programmer, you can still make use of BASIC in your computing application, and perhaps become a computer programmer through your experiences with BASIC.

BASIC is for beginners, but it can perform serious computing tasks and is the most widely used language for personal computer programming. BASIC is used for writing expensive business programs as well as games and educational programs.

There are thousands of inexpensive BASIC programs available, in books and magazines, on disks, and through user networks and electronic bulletin boards. Their variety is boundless, from space games and backgammon to artificial intelligence and decision analysis. However, to run the programs you must learn a little bit about BASIC, and the manual that's usually supplied is not much help for beginners.

As computer manufacturers developed different computers, different versions of BASIC evolved. MBASIC is the version of BASIC most CP/M computers use. MBASIC is short for Microsoft BASIC-80 and is available for nearly every CP/M system.

Typesetting Conventions

In this tutorial we use `this typeface` to show commands and program statements you type at your keyboard, and `this typeface` to show displayed messages.

In addition, we use these symbols:

↵	Return key (CR or ENTER on some keyboards)
^	Control key (CNTL, CTRL or ALT on some keyboards)

by Michael Bradley

There are lots of "dialects" of Microsoft's BASIC, including MBASIC for CP/M systems. For example, the IBM PC BASIC is an improved version of the original Microsoft BASIC; the Applesoft BASIC is an earlier version. Each works a little differently, and some, like the Apple version, are quite different.

This article is about Microsoft BASIC-80 on CP/M systems. If you have another version of CP/M BASIC, or a PC-DOS or MS-DOS version, be sure to double-check everything in this article against your own manual. In fact, double-check the article even if you do have Microsoft BASIC-80, since there are several versions of this one version of BASIC!

BASIC was created as an introductory programming language. It wasn't intended as the ubiquitous tool it's become. That's one of the main reasons BASIC programs aren't as slick as others — BASIC wasn't designed to be slick in the first place. It was designed to be easy to learn, which it is. However, because of the simplicity of the language, BASIC is not ideal for many programming problems, and in some cases where structured programming techniques are the best to use, BASIC is clumsy compared to languages like Pascal and "C".

In spite of its clumsiness, BASIC is still popular, largely because program development costs are very low. You can write, "debug" (fix errors) and publish BASIC programs quickly and inexpensively. For that reason, it is possible to find a suitable inexpensive BASIC program to handle an application, rather than spend more money on a more sophisticated program.

MBASIC and CP/M

What makes MBASIC programs different from other programs for CP/M systems is the way you execute them. Unlike programs whose filenames have the extension ".COM", an MBASIC program's filename has the extension ".BAS". And unlike a ".COM" program that you execute by simply typing its name, an MBASIC program cannot be executed directly from the CP/M command line. Though there are exceptions, normally you have to execute MBASIC.COM first.

MBASIC.COM is a CP/M program that *interprets* the MBASIC language instructions you put into a ".BAS" file. MBASIC.COM is called an *interpreter*.

When you use a computer, you create a sort of layer cake of programs in the computer's memory. The bottom layer is the operating system, CP/M. On top of the CP/M layer is the program you're executing — say a word processing program such as WordStar. The word processing program is supported by the lower layer (CP/M) so that it can use CP/M's functions of reading files on disk, displaying data on the screen, and other operating system tasks.

However, with MBASIC programs, the cake has another layer, sandwiched between CP/M and the executing program. That layer is the MBASIC.COM interpreter. BASIC programs can't use CP/M functions directly like other programs do, but MBASIC.COM can. That's one reason why you need it — so it can intercede with CP/M on behalf of your BASIC programs.

Loading and Running

Let's say that you are using a CP/M system disk in drive A, and also on this disk is MBASIC.COM and a BASIC program called Tugboat, a game in which you get to play a tugboat captain in San Francisco Bay. The game is in a file named TUG.BAS.

You're at CP/M level and the system prompt, `A>`, is onscreen. To start the game, enter this command:

```
A>MBASIC TUG↵
```

CP/M loads MBASIC into memory, and then MBASIC loads the TUG program. Notice that you didn't have to include the file extensions ".COM" and ".BAS" in the command line. However, if Tugboat's file had a different extension, e.g., TUG.BOT, the command would have to look like this:

```
A>MBASIC TUG.BOT↵
```

You must include a BASIC program's file extension if it's *not* ".BAS".

If you keep ".COM" files in drive A and your BASIC programs in drive B, make drive B the logged drive by typing `B:`↵, then start Tugboat with this:

```
B>A:MBASIC TUG↵
```

You can start the MBASIC interpreter without automatically running an MBASIC program. For this example we assume that on the disk in drive A is MBASIC.COM and some BASIC programs, including two games, TUG.BAS and HOUSE.BAS (the latter being a game in which you build a house, from foundation to roof).

To start the MBASIC interpreter, type:

```
A>MBASIC↵
```

MBASIC's sign-on message has six lines. The first one shows how much room is left in the computer's memory, as in `23674 bytes free`. The second and third lines tell you what version of MBASIC you're using. The fourth line is a copyright notice, and the fifth is the date the program was created. The last line is the one that tells you it's "ok" to activate the MBASIC interpreter with MBASIC commands. It is called the MBASIC system prompt:

```
Ok
```

Now you're ready to run MBASIC programs, or write your own MBASIC programs. MBASIC is truly dual-purpose — you can use it to run existing programs, or as a program development tool that interprets the BASIC statements and commands you enter at your keyboard.

Having started MBASIC, you can run Tugboat with this command:

```
Ok
RUN ''TUG''↵
```

Or you can run the House game with this command:

```
Ok
RUN ''HOUSE''↵
```

The quotation marks around the file names TUG and HOUSE are necessary once you are in MBASIC. Although CP/M knows a filename when it sees one, MBASIC does not know a filename unless you surround it in single or double quotes.

Notice that you didn't have to include the extension ".BAS". MBASIC assumes that the filename has a ".BAS" extension; if your program does not have a ".BAS" extension, you must supply the extension (also within the quotes).

Now, suppose you've been playing Tugboat for a while and have become bored with it. You want to play House. You do *not* have to exit to CP/M to do it. You can stop Tugboat and start House without leaving MBASIC, in one of two ways. Try Tugboat's exit command first. Some programs will exit to MBASIC, instead of going all the way to CP/M. You'll know you're at MBASIC level if the system prompt is `Ok` instead of `A>`.

Suppose, however, that Tugboat does exit to CP/M. Well, you can get around the exit command easily enough — just don't use it. Stop Tugboat by typing Control-`C`. You can type Control-`C` any time, in any program, whenever you want to stop. Control-`C` functions very much as it does in CP/M — it stops programs cold.

When the programs stop, MBASIC displays a "break message" at the left edge of the screen:

```
Break in 1000
```

On the Screen	Control Character	What Happens
`Ok`	`^C`	Stops a program completely and displays the MBASIC system prompt `Ok`. Enter RUN, etc., to re-start from the beginning, CONT to resume where you left off, or SYSTEM to exit to CP/M.
`^O`	`^O`	Pauses a program. You can't enter data, but you can press Return, which re-starts the program, repeating the last step. Or you can enter `^O` again, which re-starts the program at the next step.
`^U`	`^U`	All this does is enter a line feed. The cursor moves down one line and sits at the far left of the screen. Continue as if nothing had happened.
`!`	`^A`	Stops a program and enters a line feed. You can't enter data. After you've pressed Return, the program picks up where it left off. Sometimes this results in losing a step.
`#`	`^X`	Just enters a line feed. Continue as if nothing had happened.

Figure 1. *MBASIC Control-key commands that display something on the screen when you enter them.*

```
Ok
```

The number in the break message is the number of the line on which the program has been stopped. You can ignore it.

To re-start the same program you stopped, just type `RUN`↵. To start a different program, type the following RUN command with your filename:

```
Ok
RUN ''filename''↵
```

(Remember to use quotation marks around the filename.) Either way, when you type RUN, the program starts at their beginning. If you're re-starting the same program you stopped, all the data you were working on will be lost.

To correct typing errors while entering data, use the same editing commands as in CP/M. Control-`H`, Backspace and Delete (or Rubout) work the same way as in CP/M. That means that if your Delete (or Rubout) key repeats rather than erases letters, it will do the same in MBASIC. In some cases it is more awkward than CP/M — the repeated characters may be displayed in reverse order and enclosed in slant bars. For example, erasing `words` will produce this display:

```
words/sdrow/
```

If you can, use Control-`H` or Backspace to erase letters, because on most systems they actually erase letters without repeating them.

After stopping a program with Control-`C`, you can exit all the way to CP/M by entering:

```
Ok
SYSTEM↵
```

Control-key Commands

While you're running programs, a few Control-key commands are available. Some of them stop a program, but you can usually re-start the program where you left off. There's not much reason for you to use most of these commands, but if you enter one accidentally, it's important to know how to get yourself out of trouble.

If a program stops in midstream, check the screen for an explanation. Look first for an error message (more about error messages later). If there isn't one, look for MBASIC's system prompt, `Ok`. If the last message on the screen is the break message, the program stopped because you entered Control-`C`.

If there's neither an error message nor a break message, you entered another Control-key command. Look for `^O`, `^U`, `!` or `#` on the last line. These symbols mean that you entered, respectively, Control-`O`, Control-`U`, Control-`A` or Control-`X`. If none of these is on-screen, you may have entered one of the other Control-key commands.

Figure 1 shows the Control-key commands that display something on the screen when you enter them.

MBASIC Commands

Starting MBASIC alone or stopping a program with Control-`C` puts you at what's called "command level." All this means is that you're not running a program at the moment — you could be writing one instead. A great many commands are available to you at the MBASIC command level. Most of them are for programming, but a few are useful for everybody. You know one already: RUN.

The MBASIC commands work whenever MBASIC's sys-

Control Character	What Happens
`^G`	If your terminal has a beeper, this makes a beep. If there's no beeper, there's no beep.
`^I`	This moves the cursor to the next tab stop, just like in CP/M. Continue as if nothing had happened.
`^M`	The same as pressing Return.
`^R`	Enters a line feed. Continue as if nothing had happened.
`^S`	If the program has paused on its own to wait for input from you, `^S` has no effect at all. But if it was busy with something, like saving a file or analyzing data, `^S` makes the program pause. Enter `^Q` to resume.
`^Q`	Has no effect unless you've paused the program with `^S`, in which case it resumes the program where you left off.

Figure 2. *MBASIC Control characters that perform functions without displaying themselves.*

tem prompt, `Ok`, is on the screen, and they *must* be followed by a Return key (shown as ⮐ in examples). The commands do *not* work while you're running a program.

Program Control Commands

Use the following commands after starting MBASIC or after stopping a program with Control-`C`:

`RUN`⮐

This re-starts a program you stopped with Control-`C`; the program starts at its beginning. If you supply a program's filename, as in `RUN ''HOUSE''`⮐, the command starts that program.

`CONT`⮐

This re-starts the program wherever it was interrupted by Control-`C`. If the program had already paused to wait for input from you, `CONT`⮐ allows you to continue as if nothing had happened.

`LOAD ''filename''`⮐

This brings a program into the computer's memory. Include the program's filename, e.g., `LOAD ''PAD''`⮐. However, the program won't start until you enter `RUN`⮐.

`RESET`⮐

If you entered Control-`C` while your program was reading or saving data files, it's a good idea to enter `RESET`⮐ just before exiting to CP/M. When a BASIC program opens a data file, the file stays open until the program closes it. If you stop the program with Control-`C` before a new file has been closed, CP/M may not add the file's name to its list of files in the disk directory. The file will then be invisible and you'll lose it. To make sure this doesn't happen, type `RESET`⮐ to close all open files.

`SYSTEM`⮐

Enter this to exit to CP/M.

Disk and File Commands

These commands work with all files, not just MBASIC ones. Be careful of using lower-case letters in file names. When you type a filename with a CP/M command, it doesn't matter if you use UPPER-case or lower-case letters, because lower-case letters are converted to UPPER-case automatically.

But in MBASIC, no such conversion takes place. If you use lower-case letters to name a file that was saved originally with capitals, e.g., `RUN ''tug''`⮐ instead of `RUN ''TUG''`⮐, you'll get an error message that says `File not found`.

Be careful with lower-case letters when your MBASIC program saves a data file. Some MBASIC programs automatically convert lower-case letters in filenames into UPPER case, but some don't. Not converting is okay as long as you're using MBASIC, but it gets you into trouble at CP/M level.

An MBASIC data file named "tug.dta" will show up in the CP/M directory as `tug.dta`, exactly as you typed it. But the name, and thus the file, will still be invisible to CP/M and other CP/M programs, because if you specify a filename in lower case letters, the letters are converted to UPPER case and therefore you cannot reference the lower-case filename.

`FILES↵`

This displays the directory of files on the disk, the way DIR does in CP/M. `FILES↵` displays the files on the logged drive. `FILES ''B:*.*''↵` displays the files on drive B. CP/M's wild card characters, ? and *, can be used in filename matches.

`KILL↵`

This command erases files, just as CP/M's ERA does. Remember to enclose the filenames in quotation marks. You can include the drive letter with the name, as in `KILL ''B:PAD.DTA''↵`. You have to include the filename extension with this command, even when it is ".BAS". If you leave it out, you'll get the error message `File not found.`

`NAME oldname AS newname↵`

This renames a file. Its format makes it easier to use than CP/M's REN command. It is okay to include a drive letter, as in `NAME ''B:TUG.BAS'' AS ''SFBAY.BAS''↵`. Here the name of Tugboat's file has been changed from TUG.BAS to SFBAY.BAS. Note that you have to include the file extension ".BAS" when using the NAME command.

`SAVE ''d:filename''↵`

This command saves the program currently in memory in a disk file — you type the filename and disk drive (`d:`).

MBASIC can save a file in three different formats. The first and default format compresses the file to save disk space:

```
SAVE ''SFBAY''↵
```

If you don't name an extension, MBASIC will automatically use ".BAS" for the file.

The second format, called ASCII (for American Standard Code for Information Interchange), stores the file in a textual format that you can view on your screen or edit with your word processing program (such as WordStar). You should use the word processing program's "non-document" mode (such as the WordStar `N` command from the startup menu) to edit a BASIC program, so that you get no special printing characters mixed in with the BASIC text.

Here's a sample command that stores SFBAY.ASC as an ASCII file (the `,A` option is used):

```
SAVE ''SFBAY.ASC'',A↵
```

The `,A` comes *after* the second quotation mark. The extension ".ASC" is the customary way of indicating that the file was saved in ASCII format.

The third format saves programs in a "protected" way. No one can display, print or save new versions of the program when it is in memory:

```
SAVE ''SFBAY'',P↵
```

Commercial BASIC programs are usually supplied in this protected format. If you try to save or list them, you'll get this error message: `Illegal function call`. This is because the ability to save a program to disk or list a program to the printer could be used to "pirate" the program (make unauthorized copies without paying).

Listing Programs

`LIST↵`

Provided the program in memory is not saved in protected format, you can display it on the screen with the LIST command. LLIST (with two "L"s) sends the program to your printer.

Terminals can display only 24 lines at a time, so `LIST↵` by itself causes the entire program to scroll by on the screen. To stop the scrolling, type Control-`S`. To start scrolling again, type Control-`S` again.

Every MBASIC line begins with a number. Program lines are always numbered, usually in a regular sequence such as 1000, 1010, 1020, 1030, etc. Since the lines are numbered, you can LIST them in chunks that fit on the screen by specifying a range of line numbers:

```
LIST 1000-1100↵
```

You can also stop a scrolling LIST with Control-`C`, as well as Control-`S`, but you'll have to enter another LIST command to re-start it.

`TRON↵` and `TROFF↵`

These commands are for "debugging" (correcting programs that have "bugs"), but try it if you're curious about what happens when a program runs. Stop the program by typing Control-`C`, type `TRON↵`, and then type `RUN↵`. As the program runs, line numbers will be displayed at the left edge of the screen. They show your progress through the program. To stop the display, stop the program with Control-`C` and type `TROFF↵`.

Errors

MBASIC programs have what are called "error-trapping routines," which are parts of the programs that catch your mistakes. There are two kinds: routines built into MBASIC itself and routines written individually for each MBASIC program. Most programmers try to keep you from seeing MBASIC's own messages, partly because MBASIC's messages are so cryptic. The messages make sense only after you've seen them a few times and wrestled with their meanings.

A more important reason for "hiding" MBASIC error messages is that they stop a program just as if you'd typed Control-`C`. Such an abrupt halt is useful for debugging purposes, but it's not good for business programs.

Therefore, programmers write their own error-trapping routines, which only pause the program to explain that an error occurred. Some messages even tell you how to fix the error, or what to type. You can then pick up where you left off with no damage done. Sometimes the routines recycle your input in such a way that you're not aware of having made an error at all.

Programmers can also add routines that check data as you enter it, to be sure it's in the necessary format. They check for inappropriate data, inappropriate menu selections, and so on.

Although there may be a chance you will never see MBASIC's own error messages, you may one day see a damaging one — for example, one about disk access — that you must know how to recover from. So, to be on the safe side, I'll explain them, including some that may turn up only once in a great while.

First, let's see how to recover from an MBASIC error message.

Recovering From Errors

You should read the message and figure out what caused it to appear. Then, if you were running a program, re-start it by entering `CONT`↵ or `RUN`↵, depending on whether you want to pick up where you left off (`CONT`↵), or go back to the beginning of the program and start over (`RUN`↵). Your decision will depend on what the message is.

In an extreme case, if neither `CONT`↵ nor `RUN`↵ work, try this:

```
NEW↵

Ok
RUN ''filename'' ↵
```

The NEW command clears out RAM. If the program's been damaged while it was running, this command sequence re-loads the undamaged version that's on the disk.

MBASIC Error Messages

Let's look at the messages' general format:

```
Out of memory in 1450

Ok
```

The typical error message states what the error is (`Out of memory`), gives the number of the line the program stopped on (`1450`), and then displays the MBASIC system prompt (`Ok`). You won't need the line number information, but you should look up the error before deciding what to do next.

Disk and File Errors

`Bad file mode`

You probably tried to LOAD a special kind of data file, one that was created in a mode that won't allow you to load it. There's nothing you can do. You just can't load the file.

`Bad file name`

When you named a file, either in a program or in a SAVE command, you didn't follow CP/M's file naming rules (up to eight primary name characters with optional period and three-character extension). Enter the name again. Enter `CONT`↵ first if you're running a program.

`Disk full`

You tried to save a file, but there's not room on the disk for it. If your computer has a second drive, try saving the file there. Enter `CONT`↵ first if you're running a program.

NOTE

Be careful of file space when you're using MBASIC. If you get a disk full error, you *cannot* replace a full disk with another one. CP/M will give the new disk read/only status, and when you try to save the file, CP/M will boot the disk, pushing MBASIC, your program and your data out of RAM. PC-DOS and MS-DOS don't do this, so you don't have to be so careful.

`Disk I/O error`

Somehow, during a disk access procedure, e.g., reading or writing a file, MBASIC or CP/M just blew it. CP/M will re-start, clearing everything from RAM, and there's nothing you can do about it.

`File already exists`

This occurs only when you're renaming a file with the NAME command. It means the *new* name you entered is the name of another file on the disk. Enter a different new name.

`File already open`

You're trying to do something with a file that's been left open (see the RESET command for what "open" means). The file will have to be closed. You could enter `CLOSE`↵, which closes all open files. If this occurred because you decided to continue a program using `CONT`↵, you probably have to re-start the program at the beginning by typing `RUN`, rather than picking up where you left off with `CONT`↵.

`File not found`

The file you're looking for isn't where you thought it was. Check your typing of the name. If it's okay, the file may be on another disk, it may have been erased, or you may have renamed it. You may get this message even if you're not looking for a file — your program could be looking for it. Get a copy of the missing file from the distribution or backup disk, if you can.

`Internal error`

Good luck! MBASIC has gone screwy. You may have damaged the disk or erased part of MBASIC.COM somehow. Get a copy of MBASIC.COM from the distribution or backup disk. Microsoft would appreciate your letting them know what happened, unless you're sure it's your own error.

Program Errors

`Can't continue`

After a program was stopped by an MBASIC error message, or by you typing Control-`C`, you tried to re-start it, but it won't re-start. The error may be on a line that prevents the program from resuming. If you stopped the program yourself with Control-`C`, did you fool around with it, or even erase it? Try clearing memory with `NEW`↵, and then RUN a fresh copy from disk.

`Illegal direct`

You've been experimenting. You entered a command that should appear only within a program. Nothing serious has happened, but you're under arrest!

`Illegal function call`

See the discussion of the SAVE command.

`Line buffer overflow`

You tried to enter a line of data that's too long for MBASIC to digest, i.e, it has more than 255 characters. Ordinarily, programs screen your input for this sort of thing. (You can get around the screening, though. For example, when you're asked to type a `Y` or `N` for "yes" or "no," you can type 256 `Y`'s.) To continue the program, type `CONT`↵.

`Out of memory`

RAM is completely full. This may be the programmer's error, or yours. The programmer should foresee how much memory your program will need, even in the most extreme cases. However, if you're using something like a keyboard re-definition program or a CP/M menu shell, the responsibility is probably yours. These programs take up space your BASIC program apparently needs.

For example, I normally use SmartKey, a keyboard re-definition program, but not when I run the program I wrote myself. My program needs the RAM space for its calculations.

To clear the other program from memory, you'll have to exit all the way to CP/M by typing `SYSTEM`↵.

`Overflow`

The result of a calculation is too big. This is likely to be the programmer's error. There's not much you can do about it, except start over using RUN. Check whether the data you're entering is correct.

`? Redo from start`

This is a unique error message. The line number isn't displayed, nor is the system prompt, `Ok`. The message means you entered a particular character MBASIC won't accept. Maybe it's a letter when the program will accept only numbers. Maybe it's a comma — MBASIC never accepts commas.

Just retype the one line you were typing when this message interrupted you — without the offending character, of course.

`Syntax error`

You've been experimenting again! Well, enjoy yourself. There's no harm done. You can get this message, with a line lumber of `65535`, when you stop a program with Control-`C` and then mistype a direct command like RUN or SYSTEM. But there is no line 65535. This is just MBASIC's clumsy way of asking, "Say what?" Don't worry about it — just enter the command correctly.

`Unprintable error`

MBASIC is not swearing at you. It's just that the programmer defined something as an error, but then forgot to include instructions in the program for handling the error. Try to figure out what you did, and don't do it again!

Exceptions

As I said in the beginning of the article, you normally have to start the MBASIC interpreter before loading your program. Well, this isn't always the case. Some programs have shortcuts that start the interpreter for you.

One shortcut involves putting a boot subroutine at the beginning of the program. The subroutine's activated when you boot CP/M, loading MBASIC and your program for you. Another shortcut is an autostart program, a .COM file that works pretty much like the boot subroutine, except that you can activate it without resetting the computer. All you do is enter the .COM file's name.

Another exception is when programs are stored on the distribution disk as system files. In CP/M, a system file isn't listed in the directory, so you won't even know it's on the disk. The file that is shown will be a special program that copies the "real" program to your working disk. However, unlike the protect option in the SAVE command, you can remove a file's system status, *if* you figure out the file's name. At CP/M level, copy STAT.COM to the MBASIC program disk, then enter this:

```
A>STAT filename $DIR↵
```

Even though your BASIC programs may load automatically and be saved as system files, you can use all the commands explained in this article.

Want to Know More?

The MBASIC manual most people get with their computers is the reference manual supplied by Microsoft. In the back of the manual is a separate appendix for every version of Microsoft BASIC-80. Look up your version, and check for differences with things I've said here in the article.

What do you do if you want to learn more about programming? Here's what I did.

First, I bought Jerald Brown's great introduction, *Instant Basic, 2nd Astounding! Edition*, published by dilithium Press. I love this book. It's fun, it's smart and it works.

Next I found a book of program listings. There are lots of them on the market. Find one with programs that pique your interest. I copied a few from mine and then started modifying them, making them easier to use — friendlier and faster — and adding features and capabilities. I made my share of mistakes and had a great time.

Finally, if you get really serious about programming, you might try these books:

BASIC Programming Self-Taught, by Seymour Hirsch (Reston).

Microcomputer Software Design, by Sally Campbell (Prentice-Hall).

Microsoft Basic, 2nd ed., by Ken Knecht (dilithium Press).

Standard BASIC Programming for Business and Management Applications, by James Quasney & John Maniotes (Boyd & Fraser).

Also, check the *Subroutine Sandwich* books. They list often-used subroutines, like conversions, geometric and trigonometric calculations, interest, etc.

Computer experts I know tell me the better programs aren't necessarily written by technically superior programmers. They're written by people who start with an idea of how a program should run on the screen. Writing the program becomes, then, a matter of figuring out how to use the capabilities of the programming language to get exactly what you want.

One old pro had a different slant. He told me the best way to write a program is to do a good flowchart and a description of your system requirements, and then pay a professional programmer to do the rest. That's probably excellent advice, but not much fun.

Micro Applications & Hardware

CP/M Plus
for Cromemco
- More than 60K for user-programs in banked environment.
- Breakthrough in memory management.
- Fast efficient disk operation.
- Support for 1-16 drives of up to 512 megabytes each, including hard disks.
- All the features of CP/M 2.2... plus more!

Expand CDOS Emulator
for Cromemco
- Take advantage of large program to run Cromemco software under CP/M Plus or CP/M 2.2 or MP/M
- Specially structured basic programs

CP/M 2.2
for Cromemco
- Industry standard implementation.
- Multiple disk formats compatible with 51/4 or 8" drives.
- Fast diskette backup with our copy program.
- Most source code provided
- Full documentation.

MP/M II
for Cromemco
- Multi-user, multi-tasking operating sytem.
- Upward compatible.

Select™ Word Processor $229.
Four Powerful Programs in one System
- **Editor**—single key commands.
- **Superspell**—automatic spelling correction.
- **Teach**—on screen tutorial.
- **MergePrint**—personalized form letters done easily.

We also sell hard disk systems

MICAH

(415) 331-6422 15 Princess St., Sausalito, CA 94965

Circle number 11 on the reader service card.

QUALITY SOFTWARE AT REASONABLE PRICES
CP/M Software by
Poor Person Software

Poor Person's Spooler $49.95
All the function of a hardware print buffer at a fraction of the cost. Keyboard control. Spools and prints simultaneously.

Poor Person's Spread Sheet $29.95
Flexible screen formats and BASIC-like language. Preprogrammed applications include Real Estate Evaluation.

Poor Person's Spelling Checker $29.95
Simple and fast! 33,000 word dictionary. Checks any CP/M text file.

aMAZEing Game $29.95
Arcade action for CP/M! Evade goblins and collect treasure.

Crossword Game $39.95
Teach spelling and build vocabulary. Fun and challenging.

Mailing Label Printer $29.95
Select and print labels in many formats.

Window System $29.95
Application control of independent virtual screens.

All products require 56k CP/M 2.2 and are available on 8" IBM and 5" Northstar formats, other 5" formats add $5 handling charge. California residents include sales tax.

Poor Person Software
3721 Starr King Circle
Palo Alto, CA 94306
tel 415-493-3735
CP/M is a registered trademark of Digital Research

Circle number 14 on the reader service card.

CP/M-80 •
PC-DOS •
MS-DOS •

Giving Data To a BASIC Program From the CP/M Command Line

Before starting a CBASIC program to select an option and provide data, try this method of giving data to the program with the command you use to run the program.

How many times have you wanted to run a program and supply simple data, say a person's name, or a program option, on the same command line with the program name? Let me explain. Suppose you have a program called ADDRESS that has a list of names and addresses in it. You can call up the program, select the menu entry that says "Display Addresses," type in the name of your data file, enter the name of the person whose phone number you want to see, and then let your program do the work. You call up the program by typing its name at the CP/M `A>` command prompt:

```
A>ADDRESS ↩
```

Typesetting Conventions

In this article we use `this typeface` to show commands and expressions you type at your keyboard, and `this typeface` to show displayed messages.

In addition, we use ↩ for the Return key (CR or ENTER on some keyboards), and ^ for the Control key (CNTL, CTRL or ALT on some keyboards).

by David M. Alexander

David M. Alexander is an attorney who writes articles for various computer magazines. He uses the method described in this article to make system operations easier.

You can then pick the appropriate data filename and provide the customer's last name to display the customer's address.

Would it not be easier if you could type both the data file name and the name you want to look up on the same line with the program name?

```
A>ADDRESS FRIENDS SMITH ↩
```

One of the modules in a BASIC professional billing and invoicing program is called ADDBILL.COM. It accesses the billing addresses. I wanted to be able to use it like a Rolodex®, that is, be able to type the following:

```
A>ADDBILL DATAFILE SMITH ↩
```

This would display a Smith, an address and a phone number in a matter of seconds. In this example DATAFILE is the name of the file which the program ADDBILL will look at to find the client's name, address and phone number.

I saw an article in a computer magazine describing how to do this, but I didn't like the code. I wrote my own routine in BASIC for getting information directly from the CP/M `A>` prompt.

CP/M stores the characters following a program name (e.g. ADDBILL.COM) at decimal memory location 128. All references here are in decimal. (If God had wanted us to use hexadecimal, S/He would have given use sixteen fingers and sixteen toes.)

NOTE

If you are programming in MS-DOS, then, if DEF SEG=0, this string is stored at decimal memory location 12528. The example and BASIC code in this article assume you are using a CP/M system; however, MS-DOS information is included in the code.

When you see the `A>`, `B>` or the like on the screen, it means that CP/M is ready to receive your orders. If you type in the name of a program and then press the Return key, CP/M will try to run the program whose name you have entered. To CP/M, any filename that ends with the characters ".COM" (short for COMmand) is a program file, not a data file.

For example, if you have WordStar on your disk under the filename WS.COM, you would type `WS`↵, and CP/M would look for WS.COM. If CP/M finds it, not only would CP/M run WS.COM but it would also put into the computer's internal memory (RAM) all of the characters that you typed in between the letters `WS` and the Return key.

If you typed in `WS TEXTFILE`↵, CP/M would automatically put the number 9 (the number of characters between `WS` and the Return key) at memory location 128. Remember that CP/M counts the space between `WS` and `TEXTFILE` as one character making the length 9, not 8.

Next, at location 129, CP/M would store the number 32 which is the ASCII code for the space between `WS` and `TEXTFILE`. At location 130 it would store the number 84, which is the ASCII code for the letter `T`. At location 131 it would store the number 69, which is the ASCII code for a capital `E`, and so on.

The maximum length of the string which can be entered after a command or program name is 128 minus the length of your command or program name — in this case 128-2 (`WS`) = 126 minus one (for the space between `WS` and `TEXTFILE`) leaves a maximum of 125 characters which you can pass to your program.

The BASIC PEEK command looks at (peeks at) the indicated memory location and returns the number that is in that location. You then convert that number to a letter with the BASIC CHR$ function.

To re-cap, the first character that CP/M puts at location 128 is a number which tells us how many characters were typed in after the command or program name. Thus, if we typed `ADDBILL DATAFILE SMITH`, the number would be 15.

BASIC Code

Now, let's take a look at some code you can use to extract this information from the CP/M command line and directly incorporate that information into your own programs:

```
5 REM FOR MS-DOS PROGRAMS INCLUDE 'DEF SEG=0' HERE
10 gosub 3200
20 ON FLAG19% GOTO 30,100,120
30 PRINT,''*** MENU OF PROGRAM CHOICES ***''
.
.
.
100 LINE INPUT '' WHAT IS THE NAME YOU WANT ME TO FIND'';NNAME$
.
.
.
120 REM BEGIN THE BINARY SEARCH TO FIND NNAME$
         IN YOUR DATA FILE BLFLE$
3190 REM FOR MS-DOS LET POINTER%=12528 NOT 128
3200 PT%=0:POINTER%=128:LGTH%=PEEK(POINTER%):
         POINTER%=POINTER%+1:STRG$=''''
3210 IF LGTH%>26 THEN
         PRINT,''*** INPUT DATA LINE TOO LONG ***'':
         FLAG19%=1:RETURN
3220 IF LGTH%<2 THEN FLAG19%=1:RETURN
3230 FOR LOOPER%=1 TO LGTH%
3240 CHAR%=PEEK(POINTER%):CHAR$=CHR$(CHAR%):
         IF CHAR%=32 THEN GOTO 3320
3250 STRG$=STRG$+CHAR$:POINTER%=POINTER%+1
3260 NEXT LOOPER%:IF LEN(BLFLE$)<1 THEN GOTO 3300
3270 IF LEN(STRG$)<1 THEN GOTO 3305 ELSE NNAME$=STRG$:
         FLAG19%=3:RETURN
3300 IF LEN(STRG$)>0 THEN BLFLE$=STRG$ ELSE
         FLAG19%=1:RETURN
3305 FLAG19%=2:RETURN
3310 BLFLE$=STRG$:STRG$='''':PT%=2:GOTO 3260
3320 POINTER%=POINTER%+1:
         IF LEN(STRG$)<1 THEN GOTO 3260
3330 IF PT%=2 THEN GOTO 3260
3340 GOTO 3310
```

YOU DON'T NEED
EUREKA!™ ??
CONGRATULATIONS!!

We admire your talents. After all, few people can remember where to find that six month old letter to Wonder Waffle Works, or which of the twenty versions of IMPORTNT.BAS is the one you need yesterday.

Or maybe we should envy your spare time. Ah, to be able to haul out a stack of disks, slip each one into a drive, browse through the directory, and TYPE the various prospects to find that one file or program.

Or perhaps you're the adventurous type who thrills to the challenge of groping through scantily labeled disks, cheering that magical moment when hidden treasures are uncovered.

On the other hand, it occurs to us that you just may not know the advantages of **EUREKA!**, the fast, menu driven disk cataloger for CP/M. **EUREKA!** puts your entire disk library at your fingertips. Files may be found quickly and easily - by name or by comments you can put in the file itself. Of course the manual includes a tutorial to help you get started.

Still only $50. Ask your dealer, or contact:

MENDOCINO SOFTWARE COMPANY, INC.

Dept U-1,
P.O. Box 1564
Willits, CA 95490
Phone: (707) 459-9130

I found it!

Add $2.50 Shipping:
Calif. residents add $3.00 sales tax.

We accept VISA
& Mastercharge

CP/M is a registered trademark of Digital Research Corp.
EUREKA! is a trademark of Mendocino Software Co., Inc.

Circle number 10 on the reader service card.

At line 3200 we set POINTER% to 128, and then we PEEK at 128. The result of that PEEK is the number 15, and we let LGTH% (length of the string) equal that number (15). Next we increment POINTER% so that we will look next at the first character in the string.

At line 3210 we put a limit on the length of the string. In this case I chose the length 26 (space, 8 characters, period, 3 characters, space, 8 characters, period, three characters). If the user goes crazy and types a string longer than we can use, we find out right here.

Line 3220 checks to see if any string was typed in. If LGTH% is 0 or 1 (1 being a space following the program name), then no string was typed in and our routine will send the user to the normal beginning of the program; that is, our first program menu. FLAG19% is set to one, telling the program that nothing was typed in.

At line 3230 we have determined that a string of an acceptable length was typed in after the program name and we are going to look at it character by character. POINTER% is now set to 129 and at line 3240 we PEEK memory location 129. There is a number here which is the ASCII code of a character. We call this number CHAR%. We translate this number to a letter by letting CHAR$=CHR$(CHAR%).

At line 3240 we look to see if this first character is a space (ASCII 32). If it is a space we go to line 3320 where we skip the space by incrementing POINTER% to the next character in the string.

We want to get rid of leading characters, that is all those spaces in `ADDBILL    DATAFILE SMITH`. If the string STRG$ is less than 1 (meaning we have not yet hit even one acceptable character), then this is a leading space and we go to 3260 where we do NEXT LOOPER%. Since we have already incremented POINTER% to 130 at 3320, line 3240 will get the next character at memory location 130.

What does line 3330 do? Okay, we want to get rid of leading spaces, but we want to differentiate between a leading space and a space between words. All leading spaces are killed because they occur while the length of STRG$ is less than one.

The first time we encounter a space at the end of word, STRG$ will be equal to or greater than one (because we have hit some non-space characters already) and PT% will not yet have been set to 2. Thus we will go past 3330 to 3340 and from there to 3310 where we set BLFLE$ (BILLFILE) to STRG$. BLFLE$ is our first word; it is the billing file containing the data, the client's names and addresses.

Okay, let's go back, and let's say we are past any leading spaces. At line 3240 we set CHAR$ equal to the first letter. At line 3250 we add that letter to STRG$ and we increment the POINTER%. At 3260 we go back and get the second letter then add it to STRG$. We keep doing this until we hit a space.

Now you can see that the space between `DATAFILE` and `SMITH` takes us from 3240 to 3320 through to 3340 to 3310. At 3310 we set BLFLE$ equal to "DATAFILE", our first word. The space we have just encountered is viewed as a divider between the first and second words in the command line.

We now set STRG$ to a null, and set PT% to 2 as a flag to tell us that we have found a word. We go back to 3260 and start working on the next word, `SMITH`.

When we come to the end of the loop we want to know if the user gave us both a filename and a person name or just a filename, or gave us just a string of spaces.

If we went all the way through the loop and the length of BLFLE$ was less than 1, and if the length of STRG$ was greater than 0, then we know that all the user gave us was one word, not two. The IF-THEN statement at 3260 sends us to 3300 where we set BLFLE$ equal to that first word. 3305 then sets the flag to 2 to tell the main program that we got one word, not two.

Of course, if BLFLE$ is a null and if STRG$ is a null, then the user just gave us a string of spaces, so we set FLAG19%=1 at 3300 and return.

Now, if we did get a BLFLE$, did the user give us a human name or just spaces after the data filename? At line 3270 we look to see if the length of STRG$ is less than one; that is, is there anything in STRG$? If all we got were spaces, then it would have zero length and we would GOTO 3305 where we would again set the flag to indicate that no human name was given.

If BLFLE$ and STRG$ both have some characters in them, then NNAME$ (the variable I assign to the human name) will be set to equal STRG$, and at 3270 FLAG19% will be set to 3 to tell the program that I have both a data file name (BLFLE$) and a human name (NNAME$).

That's it. We skip the leading spaces, skip spaces between words, skip trailing spaces, check to see how many words we have, set our flag to tell the main program how many words we have, set our variables equal to the words we have, and return to the main part of the program.

You don't have to look for only two words. You can tinker with the program to allow it to handle several words, up to a maximum of 128 characters minus the length of the ".COM" program minus 1.

Remember, CP/M automatically changes lower case to UPPER case. Anything typed at the `A>` prompt will be seen by your program as UPPER case letters. Also, remember, that you must compile your program into a ".COM" file for this to work. Don't start BASIC and then try to LOAD your program followed by your data filename and expect this to work. BASIC doesn't know anything about this. You have to run your program from CP/M as a ".COM" file for this to work.

This routine provides experienced users of your programs with a nice way to avoid menu s and go directly to the information they want your program to process. Give it a try.

Next Time

We ran out of space, so we are postponing some articles for the next issue. The next issue will feature an application note on linking bowling centers in a network configuration, plus an evaluation of CLIP (application development tool), a comparison of EUREKA! and CATALOG (two disk cataloguing programs), more on public domain software, and a return of the Sleaze Patrol.

CP/M-80 •
PC-DOS •
MS-DOS •

File Handling Utilities

Making life easier with DISK.COM, NSWP.COM and D.COM

Access Information

D.COM is a public domain utility program available from many RCPMs (remote CP/M systems) as well as local and national CP/M user's groups. *(Issue #3 of User's Guide* has a list of RCPMs worldwide, and another list is forthcoming — eds.)

Typesetting Conventions

In this article we use `THIS typeface` to show exactly what you should type at your keyboard, with the special symbols ^ for the Control key (CTRL, CNTL or ALT on some keyboards), and ↵ for the Return key (RET, CR or Carriage Return on some keyboards). We also use `This Typeface` to show messages displayed by the system or by programs.

by Bob Wolff

Bob is the Product Manager for general ledger software at Tumbleweed Business Systems in Dallas, TX. He has a graduate degree in Accounting and is a self-taught computer user.

Dealing with computer files, whether they are text or program files, is unexciting drudgery. It is a pain in the neck to keep track of tens or hundreds of files... and which files have what attributes... and which disks are formatted and which aren't... and where files are located on the hard disk... and what text files say... and when to make back up copies... and when to erase files... and how much space is left on your hard disk and floppies... and juggling file names... and on and on.

At least six CP/M commands are necessary to perform these tasks, not to mention disciplined work habits. You have to learn dozens of command variations, some to use only occasionally. You have to resort to manuals that don't fully explain essential commands. Vendors who don't know can't offer help. Both novices and experienced computer users have these file and disk handling problems in common.

While the programs described in this article don't eliminate all these problems (no program can), they can lessen their impact. D.COM, DISK.COM and NSWP.COM together are an effective toolkit. These three programs replace CP/M commands like PIP, STAT, DIR, ERA, REN, TYPE and USER. In fact, because DISK.COM and NSWP.COM have many identical capabilities, it is more like two and a half programs replacing seven commands, with added ease of use and capabilities.

All these programs are running on my Kaypro 10. All the public domain programs were provided by Chris Jenkinson of MicroData Business Systems of Dallas. He also sold me my Kaypro 10, and has not forgotten me since. I have run the programs on both hard and floppy disks, with the same results. I assume they will run on any hard or floppy disk computer system running CP/M 2.2.

D.COM: The Directory

D.COM replaces DIR and STAT for displaying directories on multiple disks and user areas, sorting the file names and displaying file attributes.

D.COM takes only 4K of space and works on a floppy or hard disk. There are other utility programs, public and otherwise, that are similar to D.COM, among them SDIR.PRL under the MP/M operating system supported by Altos Computers and various XDIR.COM programs in the public domain. In addition, SD.COM (see sidebar) is similar to D.COM, but for use with MS-DOS. It was written by Micro Data (not associated with MicroData Business Systems above).

D.COM has both speed and versatility. D.COM does the functions of DIR and STAT to provide a directory of files on the current drive and user number, or the same user number of another drive. But unlike DIR, D.COM sorts the file names (by number then letters, or by ASCII sequence). STAT also does this, but STAT produces a single column of files, a waste of both screen space and paper if printed. D.COM lists the sorted file names and file sizes, but does it in columns — column 1 contains sorted file names, which are continued in column 2, 3 and 4. It does not list file names across the page, which are hard to read, but sorted in up to 4 columns.

Also, D.COM handles lists of file names longer than 23 lines (one screen) far better than DIR or STAT. It stops after 23 lines, then prompts you on the 24th line to press the spacebar to see the next 22 lines. No more having to use ControlS to stop a scrolling list! In fact, the 23rd line moves to line 1 of the screen when the next group of file names are listed. This feature is helpful when a long list scrolls on the screen.

When a list is longer than 23 lines, the columns are still sorted. Each screen is independent. The columns on screen 1 are sorted, as are columns on each subsequent screen. This is very useful when you are searching for a file, because you don't have to remember the last file name in a column from the previous screen. This is just one of several exceptional features (whoever wrote D.COM, thank you!).

D.COM has four parameters which separately or in combination expand the information provided about the files:

`U`	Include all DIR attribute files in all user numbers (user areas) with the file count and total space used.
`V`	Show the user number (UN) and attributes (AT) of each file, in addition to the file name and size.
`SYS`	Include files with SYS attribute in the file count and total space used.
`F`	Show only the file's name and extension.

You add the parameters to the `D` command line after a dollar sign (`$`). Thus, if I type `D $UV`↩, D.COM sees the `$` and attempts to interpret the letters following it. `U` and `V` must not have a space between them; if they did, D.COM would ignore the `V` or any other parameter after it.

The following section describes each parameter and provides and example. Here is the D.COM command by itself:

```
A>D↩
```

The result is shown in figure 1. All lists are presented in this form, unless you choose the `V` or `F` parameters. It shows the file's name, extension and space occupied on the disk.

The list includes all files which have the DIR attribute on the current user number and drive. You can choose another drive by entering `d:` (where `d` is an existing drive letter between A and P):

```
A>D d:↩
```

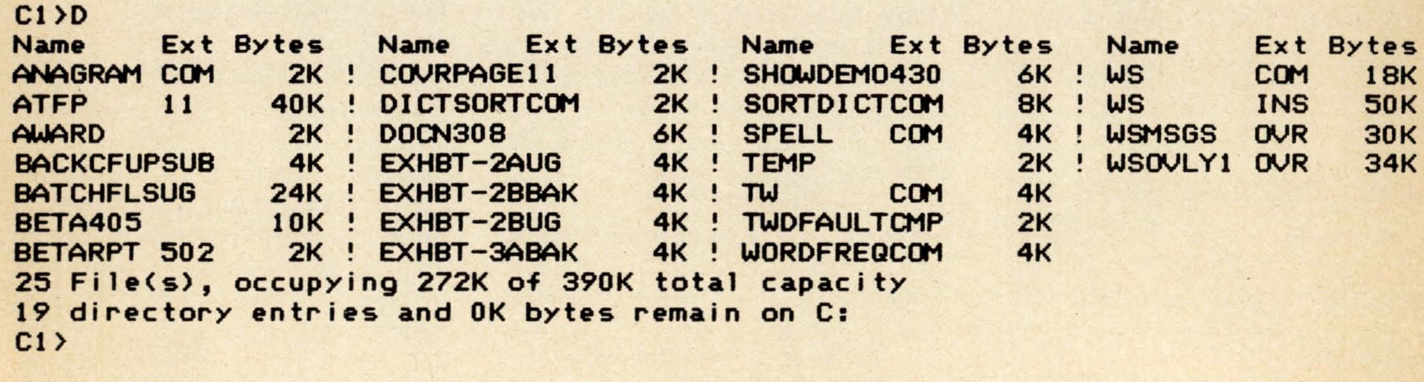
Any file with the SYS attribute, which hides a file under CP/M, is not listed or included in the file count and total space occupied by the listed files.

Note that the file names are sorted in ASCII sequence. For example, BETA405 appears before BETARPT.502 because numbers ("405") appear before letters ("RPT") in the sequence of ASCII codes *(see "ASCII Alchemy" in issue #8 — eds).* The sorted file names also appear sorted IN COLUMNS, rather than across the page. This is the more common method of presenting sorted information for noncomputer-oriented people.

Common to all directory displays are summary lines which appear after the list of file names. These lines tell you:

(1) the number of files listed,
(2) the space they occupy on the disk in comparison to its total capacity,
(3) the remaining directory entries or file names which may be added to the disk's directory, and
(4) the space remaining on the disk.

Figure 1 shows 0K left on the disk. Since the files listed occupy only 272K of the possible 390K, you may infer files are stored on other user numbers or have the SYS attribute. The `V` and



```
C1>D
Name    Ext Bytes   Name    Ext Bytes   Name    Ext Bytes   Name    Ext Bytes
ANAGRAM COM    2K ! COVRPAGE11     2K ! SHOWDEM0430    6K ! WS      COM   18K
ATFP    11    40K ! DICTSORTCOM    2K ! SORTDICTCOM    8K ! WS      INS   50K
AWARD          2K ! DOCN308        6K ! SPELL   COM    4K ! WSMSGS  OVR   30K
BACKCFUPSUB    4K ! EXHBT-2AUG     4K ! TEMP           2K ! WSOVLY1 OVR   34K
BATCHFLSUG    24K ! EXHBT-2BBAK    4K ! TW      COM    4K
BETA405       10K ! EXHBT-2BUG     4K ! TWDFAULTCMP    2K
BETARPT 502    2K ! EXHBT-3ABAK    4K ! WORDFREQCOM    4K
25 File(s), occupying 272K of 390K total capacity
19 directory entries and 0K bytes remain on C:
C1>
```

Figure 1. *The display produced by D.COM without parameters.*

SYS parameters bring out the other files residing on the disk (as shown below).

D With $U

The U parameter shows files on all user numbers:

```
A>D $U↵
```

The result of this form of D.COM is shown in figure 2. It displays in the same way as figure 1, including ASCII sequence in columns. Also, it includes only files with the DIR attribute.

In figure 1 the remaining space on the disk was 0K, and the files listed occupied only 272K of 390K total available. This indicated other files existed on another user number. In this listing all files on the disk, regardless of user number, are displayed. Yet the files listed still do not account for the total space — 384K of 390K available. This indicates one or more files with the SYS attribute are stored on the disk.

D With $V

The V parameter shows the files on the current user and drive. However, it also includes the user number (UN) where the file is stored as well as its attribute (AT):

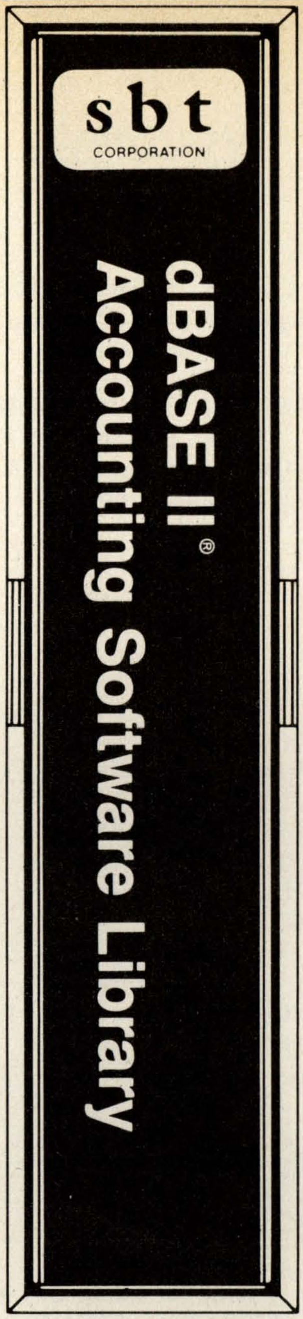
```
A>D $V↵
```

The result of this form of D.COM is shown in figure 3. Having the user number and attribute available offers useful information. For example, the attribute on TW.COM is W, meaning it is "read-only." Knowing what attributes are used is important because certain files should not be changed (if text) or erased.

As in previous listings, the file names are sorted in columns and summary information about all files is included. Again, only files with the DIR attribute are listed.

D With $UV

This is the second most useful form



SBT GIVES IT AWAY!
. . . for a very good reason — so *you* have control over your accounting. With the source code, you can add a special part number system or change a report format at will — the possibilities are endless. You get control and flexibility that's crucial to a well run business. What's more, every SBT Accounting Program can change and grow with your needs.

CHANGE IT!
With SBT Accounting Software, changes are made quickly and easily. Whether you need special invoice forms, unique mailing lists, or an inventory costing formula that would challenge an Einstein, SBT Software can be changed to fit your needs in less time than you would think!

GROW WITH IT!
When your company grows, SBT Accounting Software grows with you (right up to 65,000 customers, for example). But a system that just handles more customers is not enough. When you grow, you also change the way you do business (which means changing the way your software does business). You can make the changes yourself, or SBT will make them for you quickly and inexpensively.

INTEGRATE IT!
SBT is *the source* for modular, integrated accounting systems that can be quickly and easily changed to fit your needs. Start with the modules you need now, and add the others later. Best of all, SBT Software works on all personal computers that can run dBase II*.

.dOrder II	Sales Order Processing	$195
.dInvoice II	Billing and Inventory	$195
.dStatement II	Accounts Receivable	$95
.dPurchase II	Purchase Orders	$195
.dPayable II	Accounts Payable	$295
.dPayroll II	Payroll and Labor Acctg.	$395
.dLedger II	General Ledger/Finance	$395
.dAssets II	Fixed Asset and Depreciation Acctg.	$195
.dBackup II	Automatic File Backup	$39

For more information call SBT at 408/980-8880.

DEMO DISKS AVAILABLE

SBT Corporation
1140 Mountain View–Alviso Road
Sunnyvale, California 94089

MAY THE SOURCE BE WITH YOU

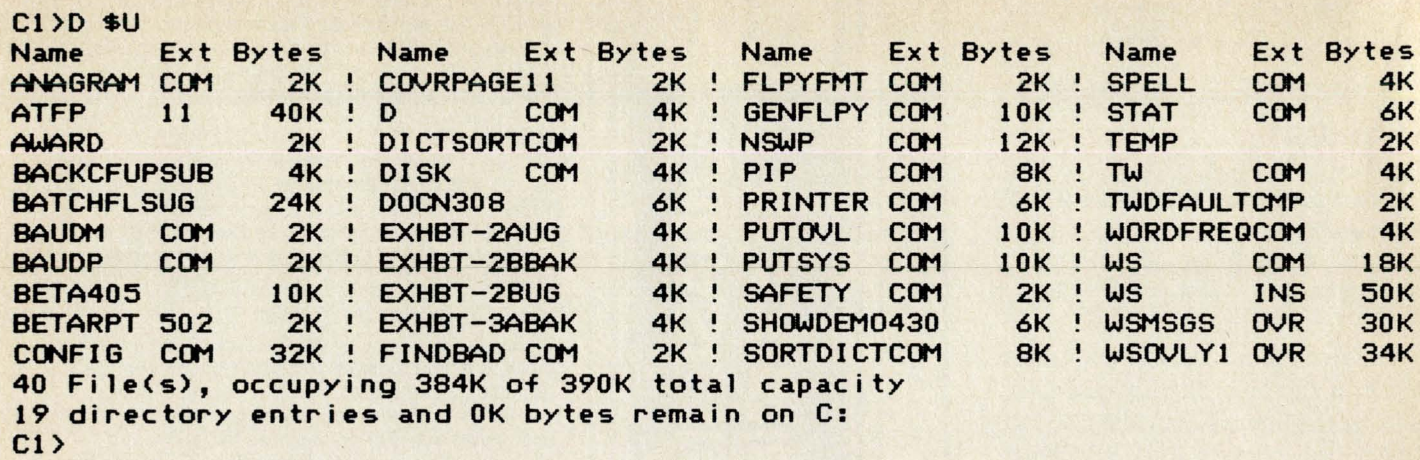
See us at booth 407 at the PC Faire!

sbt CORPORATION

dBase II* SOURCE CODE INCLUDED WITH ALL SBT ACCOUNTING SOFTWARE

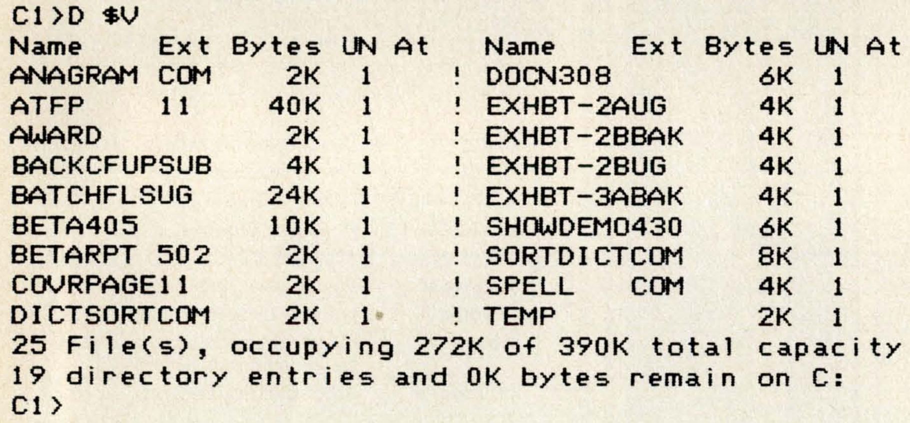
*dBase II is a registered trademark of Ashton-Tate

Circle number 16 on the reader service card.



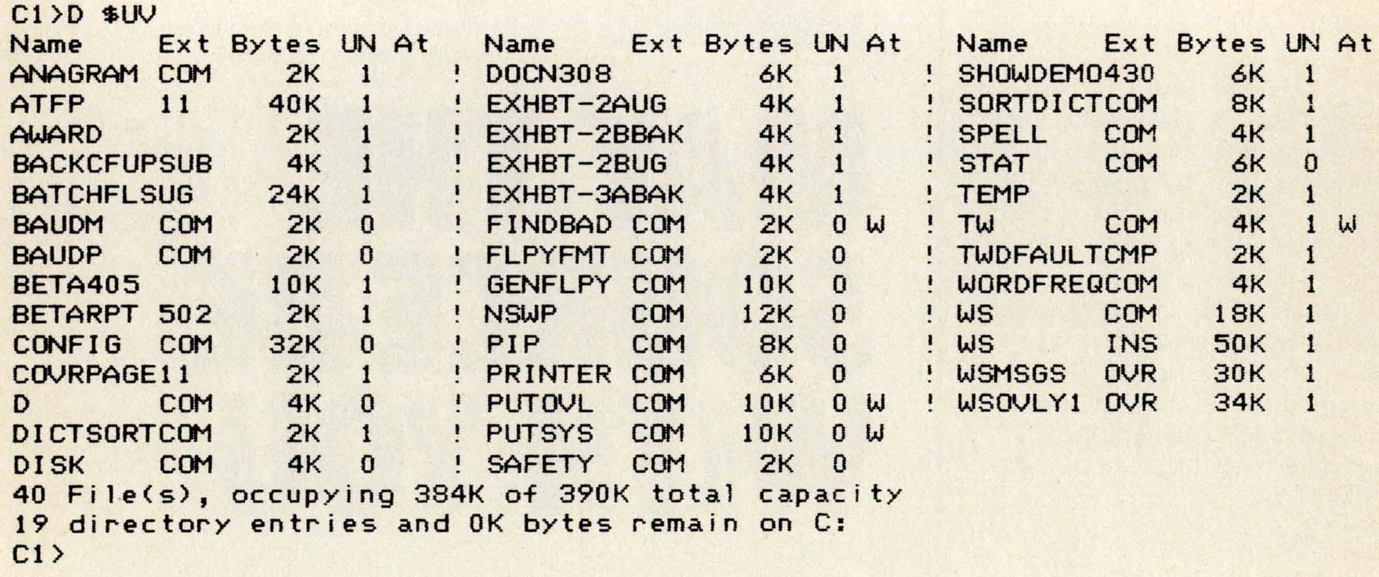
```
C1>D $U
Name     Ext Bytes   Name     Ext Bytes   Name     Ext Bytes   Name     Ext Bytes
ANAGRAM  COM    2K ! COVRPAGE11      2K ! FLPYFMT  COM    2K ! SPELL    COM    4K
ATFP     11    40K ! D        COM    4K ! GENFLPY  COM   10K ! STAT     COM    6K
AWARD           2K ! DICTSORTCOM     2K ! NSWP     COM   12K ! TEMP            2K
BACKCFUPSUB     4K ! DISK     COM    4K ! PIP      COM    8K ! TW       COM    4K
BATCHFLSUG     24K ! DOCN308         6K ! PRINTER  COM    6K ! TWDFAULTCMP     2K
BAUDM    COM    2K ! EXHBT-2AUG      4K ! PUTOVL   COM   10K ! WORDFREQCOM     4K
BAUDP    COM    2K ! EXHBT-2BBAK     4K ! PUTSYS   COM   10K ! WS       COM   18K
BETA405        10K ! EXHBT-2BUG      4K ! SAFETY   COM    2K ! WS       INS   50K
BETARPT  502    2K ! EXHBT-3ABAK     4K ! SHOWDEM0430     6K ! WSMSGS   OVR   30K
CONFIG   COM   32K ! FINDBAD  COM    2K ! SORTDICTCOM     8K ! WSOVLY1  OVR   34K
40 File(s), occupying 384K of 390K total capacity
19 directory entries and 0K bytes remain on C:
C1>
```

Figure 2. *The display produced by D.COM with the $U parameter.*



```
C1>D $V
Name     Ext Bytes UN At   Name     Ext Bytes UN At   Name     Ext Bytes UN At
ANAGRAM  COM    2K  1    ! DOCN308         6K  1    ! TW       COM    4K  1 W
ATFP     11    40K  1    ! EXHBT-2AUG      4K  1    ! TWDFAULTCMP     2K  1
AWARD           2K  1    ! EXHBT-2BBAK     4K  1    ! WORDFREQCOM     4K  1
BACKCFUPSUB     4K  1    ! EXHBT-2BUG      4K  1    ! WS       COM   18K  1
BATCHFLSUG     24K  1    ! EXHBT-3ABAK     4K  1    ! WS       INS   50K  1
BETA405        10K  1    ! SHOWDEM0430     6K  1    ! WSMSGS   OVR   30K  1
BETARPT  502    2K  1    ! SORTDICTCOM     8K  1    ! WSOVLY1  OVR   34K  1
COVRPAGE11      2K  1    ! SPELL    COM    4K  1
DICTSORTCOM     2K  1    ! TEMP            2K  1
25 File(s), occupying 272K of 390K total capacity
19 directory entries and 0K bytes remain on C:
C1>
```

Figure 3. *The display produced by D.COM with the $V parameter, showing the user area number (UN) and attributes (At).*



```
C1>D $UV
Name     Ext Bytes UN At   Name     Ext Bytes UN At   Name     Ext Bytes UN At
ANAGRAM  COM    2K  1    ! DOCN308         6K  1    ! SHOWDEM0430     6K  1
ATFP     11    40K  1    ! EXHBT-2AUG      4K  1    ! SORTDICTCOM     8K  1
AWARD           2K  1    ! EXHBT-2BBAK     4K  1    ! SPELL    COM    4K  1
BACKCFUPSUB     4K  1    ! EXHBT-2BUG      4K  1    ! STAT     COM    6K  0
BATCHFLSUG     24K  1    ! EXHBT-3ABAK     4K  1    ! TEMP            2K  1
BAUDM    COM    2K  0    ! FINDBAD  COM    2K  0 W  ! TW       COM    4K  1 W
BAUDP    COM    2K  0    ! FLPYFMT  COM    2K  0    ! TWDFAULTCMP     2K  1
BETA405        10K  1    ! GENFLPY  COM   10K  0    ! WORDFREQCOM     4K  1
BETARPT  502    2K  1    ! NSWP     COM   12K  0    ! WS       COM   18K  1
CONFIG   COM   32K  0    ! PIP      COM    8K  0    ! WS       INS   50K  1
COVRPAGE11      2K  1    ! PRINTER  COM    6K  0    ! WSMSGS   OVR   30K  1
D        COM    4K  0    ! PUTOVL   COM   10K  0 W  ! WSOVLY1  OVR   34K  1
DICTSORTCOM     2K  1    ! PUTSYS   COM   10K  0 W
DISK     COM    4K  0    ! SAFETY   COM    2K  0
40 File(s), occupying 384K of 390K total capacity
19 directory entries and 0K bytes remain on C:
C1>
```

Figure 4. *The display produced by D.COM with the $UV parameters, showing the user area number (UN) and attributes (At) for all DIR-attribute files in all user areas of the current drive.*

of D.COM:

```
A>D $UV↵
```

The result of this form of D.COM is shown in figure 4. The display includes user area numbers and attributes for each file.

The `U` and `V` parameters make D.COM search all user numbers on the current drive for the file(s) you want. For example, to search for all files with the ".TXT" extension, you type:

```
A>D *.TXT $UV↵
```

You can have D.COM search other drives by giving the drive letter before the file name. For example, if files with the extension ".TXT" weren't found on the current drive, drive C could be searched by typing:

```
A>D C:*.TXT $UV↵
```

D With $SYS

This version of D.COM produces the display shown in figure 5:

```
A>D $SYS↵
```

Compare the display in figure 5 with figure 1. This form shows one more file, WC.COM. This is the file with a SYS attribute hidden from normal viewing in the figure 1 list.

Like previous listings this form displays files sorted in ASCII sequence, with file sizes and summary information.

D With $UVSYS

If you use all three parameters you get a display like the one shown in figure 6:

```
C1>D $SYS
Name     Ext Bytes   Name     Ext Bytes   Name     Ext Bytes   Name     Ext Bytes
ANAGRAM  COM    2K ! COVRPAGE11     2K ! SHOWDEMO430    6K ! WORDFREQCOM    4K
ATFP     11    40K ! DICTSORTCOM    2K ! SORTDICTCOM    8K ! WS       COM   18K
AWARD           2K ! DOCN308        6K ! SPELL    COM   4K ! WS       INS   50K
BACKCFUPSUB     4K ! EXHBT-2AUG     4K ! TEMP           2K ! WSMSGS   OVR   30K
BATCHFLSUG     24K ! EXHBT-2BBAK    4K ! TW       COM   4K ! WSOVLY1  OVR   34K
BETA405        10K ! EXHBT-2BUG     4K ! TWDFAULTCMP    2K
BETARPT  502    2K ! EXHBT-3ABAK    4K ! WC       COM   2K
26 File(s), occupying 274K of 390K total capacity
19 directory entries and 0K bytes remain on C:
C1>
```

Figure 5. *The display produced by D.COM with the $SYS parameter.*

```
C1>D $UVSYS
Name     Ext Bytes UN At   Name     Ext Bytes UN At   Name     Ext Bytes UN At
ANAGRAM  COM    2K  1    ! DISK     COM    4K  0    ! SAFETY   COM    2K  0
ATFP     11    40K  1    ! DOCN308         6K  1    ! SHOWDEMO430     6K  1
AWARD           2K  1    ! EXHBT-2AUG      4K  1    ! SORTDICTCOM     8K  1
BACKCFUPSUB     4K  1    ! EXHBT-2BBAK     4K  1    ! SPELL    COM    4K  1
BATCHFLSUG     24K  1    ! EXHBT-2BUG      4K  1    ! STAT     COM    6K  0
BAUDM    COM    2K  0    ! EXHBT-3ABAK     4K  1    ! TEMP            2K  1
BAUDP    COM    2K  0    ! FINDBAD  COM    2K  0 W  ! TW       COM    4K  1 W
BETA405        10K  1    ! FLPYFMT  COM    2K  0    ! TWDFAULTCMP     2K  1
BETARPT  502    2K  1    ! GENFLPY  COM   10K  0    ! WC       COM    2K  1  S
CLIENT   BCN    4K  0  S ! NSWP     COM   12K  0    ! WORDFREQCOM     4K  1
CONFIG   COM   32K  0    ! PIP      COM    8K  0    ! WS       COM   18K  1
COVRPAGE11      2K  1    ! PRINTER  COM    6K  0    ! WS       INS   50K  1
D        COM    4K  0    ! PUTOVL   COM   10K  0 W  ! WSMSGS   OVR   30K  1
DICTSORTCOM     2K  1    ! PUTSYS   COM   10K  0 W  ! WSOVLY1  OVR   34K  1
42 File(s), occupying 390K of 390K total capacity
19 directory entries and 0K bytes remain on C:
C1>
```

Figure 6. *The display produced by D.COM with the $UVSYS parameters, showing the user area number (UN) and attributes (At) for all files in all user areas of the current drive, including SYS-attribute files.*

```
A>D $UVSYS↵
```

This is a Full House. It shows files from all users numbers (`U` parameter), and with attributes (`V` parameter) displayed. Also, the `SYS` parameter included files with the SYS attribute, indicated by an `S` in the `AT` column. We've now accounted for all files on the disk — 390K of 390K. Compare with figures 1 and 4.

Like the previous displays, this form displays files sorted in ASCII sequence, with file sizes and summary information.

You can probably guess what `D $USYS` and `D $VSYS` do. The first displays names and sizes of all files in all user areas of the current drive, and the second displays names, sizes, user numbers and attributes of all files in all user areas of the current drive.

D With $F

For a concise directory like the one produced by DIR, but with sorted columns and summary information, use this form:

```
A>D $F↵
```

The display is shown in figure 7. Using the `F` (just file names) parameter results in a file listing similar to the CP/M command DIR — just the file name and extension displays.

D.COM with this parameter does have advantages over CP/M's DIR alone. The files are sorted in columns while DIR just lists them as they appear in the disk's directory. Also, it summarizes the files listed while DIR does not.

In comparison to other forms of D.COM, this is not as beneficial. Entering `D`↵ for a simple D.COM display is easier than entering the `F` parameter

SD.COM For MS-DOS

SD.COM is a public domain utility program similar to D.COM. SD.COM was created to work under MS-DOS for computers like the IBM PC.

The commands to get a file listing under MS-DOS is DIR. It has several parameters which give it some flexibility, but it is essentially limited.

SD.COM was evidently written to overcome some of its limits. While SD.COM is better than DIR, it still needs to mature more to be in the same class as D.COM.

Figure 8 shows a sample file listing. Unlike DIR, it sorts the file names and puts them in 2 columns. It tells you the amount of free space as well as the current drive. And unique to most MS-DOS computers is the date and time.

Examining the list notice the sorted file names in 2 columns. This is similar to D.COM, but lacks the fine feature of sorting columns by screen. If a list of files go over 24 lines, SD.COM does not pause nor does the next sorted file appear in the second column. To me this is annoying, because D.COM has spoiled me with its pausing between screens and better sorting. Perhaps Micro Data may change SD.COM in the future.

SD.COM has several parameters just as D.COM does. They are single parameters, meaning they cannot be combined for mutual effect. Parameters are added to the basic form by entering a slash (`/`) then the parameter.

After initial testing, I got only the `/X`, `/S` and `/D` parameters to work. While this is less than half of them, these represent the most common uses.

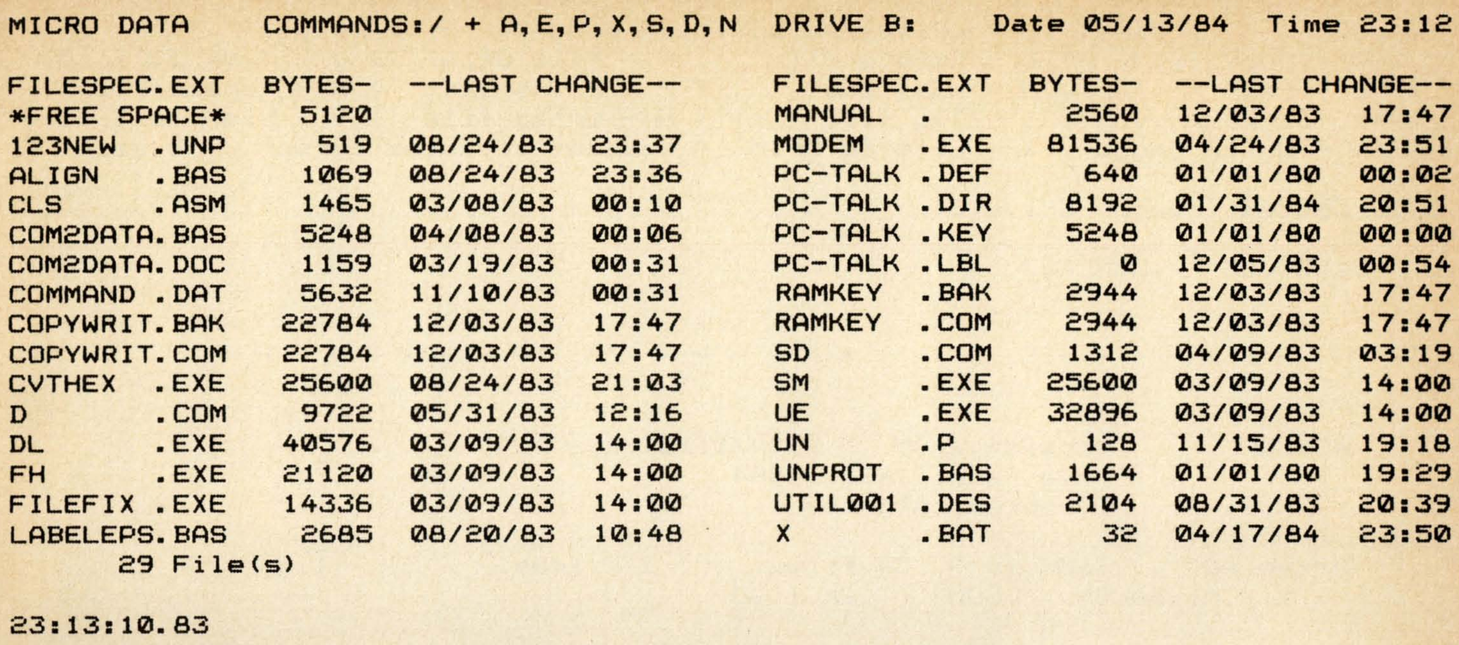
`/X` sorts by the file extension. This means the last three characters are put in order. This type of file listing would be useful to me in writing this article, because I used ".UG" as the extension for half a dozen different files. To check all of them out, `SD /X`↵ puts them all together.

`/S` sorts by the size of the file. This means the files are ordered from smallest to largest. Again, this has helped me see trends in document development as I've developed more polished versions of this article.

`/D` sorts by the date the file was last changed. MS-DOS dates and times each file when the file is created or changed. Sorted dates make it easier to determine the latest version of a document when too many similar names exist.

```
C1>D $F
Name     Ext   Name     Ext   Name     Ext   Name     Ext   Name     Ext
ANAGRAM  COM ! BETA405      ! EXHBT-2AUG   ! SORTDICTCOM ! WORDFREQCOM
ATFP     11  ! BETARPT  502 ! EXHBT-2BBAK  ! SPELL    COM ! WS       COM
AWARD        ! COVRPAGE11   ! EXHBT-2BUG   ! TEMP         ! WS       INS
BACKCFUPSUB  ! DICTSORTCOM  ! EXHBT-3ABAK  ! TW       COM ! WSMSGS   OVR
BATCHFLSUG   ! DOCN308      ! SHOWDEMO430  ! TWDFAULTCMP  ! WSOVLY1  OVR
25 File(s), occupying 272K of 390K total capacity
19 directory entries and 0K bytes remain on C:
C1>
```

Figure 7. *The display produced by D.COM with the $F parameter, showing only file names and extensions, with summary information, for DIR-attribute files.*



```
MICRO DATA     COMMANDS:/ + A,E,P,X,S,D,N  DRIVE B:     Date 05/13/84  Time 23:12

FILESPEC.EXT   BYTES-   --LAST CHANGE--            FILESPEC.EXT   BYTES-   --LAST CHANGE--
*FREE SPACE*    5120                               MANUAL   .      2560  12/03/83   17:47
123NEW   .UNP    519   08/24/83    23:37           MODEM    .EXE  81536  04/24/83   23:51
ALIGN    .BAS   1069   08/24/83    23:36           PC-TALK  .DEF    640  01/01/80   00:02
CLS      .ASM   1465   03/08/83    00:10           PC-TALK  .DIR   8192  01/31/84   20:51
COM2DATA.BAS    5248   04/08/83    00:06           PC-TALK  .KEY   5248  01/01/80   00:00
COM2DATA.DOC    1159   03/19/83    00:31           PC-TALK  .LBL      0  12/05/83   00:54
COMMAND .DAT    5632   11/10/83    00:31           RAMKEY   .BAK   2944  12/03/83   17:47
COPYWRIT.BAK   22784   12/03/83    17:47           RAMKEY   .COM   2944  12/03/83   17:47
COPYWRIT.COM   22784   12/03/83    17:47           SD       .COM   1312  04/09/83   03:19
CVTHEX   .EXE  25600   08/24/83    21:03           SM       .EXE  25600  03/09/83   14:00
D        .COM   9722   05/31/83    12:16           UE       .EXE  32896  03/09/83   14:00
DL       .EXE  40576   03/09/83    14:00           UN       .P      128  11/15/83   19:18
FH       .EXE  21120   03/09/83    14:00           UNPROT   .BAS   1664  01/01/80   19:29
FILEFIX .EXE   14336   03/09/83    14:00           UTIL001  .DES   2104  08/31/83   20:39
LABELEPS.BAS    2685   08/20/83    10:48           X        .BAT     32  04/17/84   23:50
         29 File(s)

23:13:10.83
```

Figure 8. *The display produced by SD.COM (a similar program for MS-DOS).*

with D. While D.COM without parameters may give more information than you want, for me the extra keystrokes for this parameter is more work than blocking out the file sizes.

You can combine other parameters with F, such as SYS to display SYS-attribute files, or U to display files from all user areas:

```
A>D $UFSYS↵
```

Note that some parameters override others. If you use the V parameter with F (specifying V first), the F overrides the request from V to show user numbers and attributes, and you end up with a display exactly like figure 7.

This illustrates that some combinations of parameters are illogical to D.COM. In this case, V indicated one form of display, then F told it to use another form of display. If you experiment with different combinations of parameters, D.COM does not get confused — it chooses the "logical operation" (whatever it was programmed to choose), rather than simply "bombing out." This is one indication D.COM is a well-designed program.

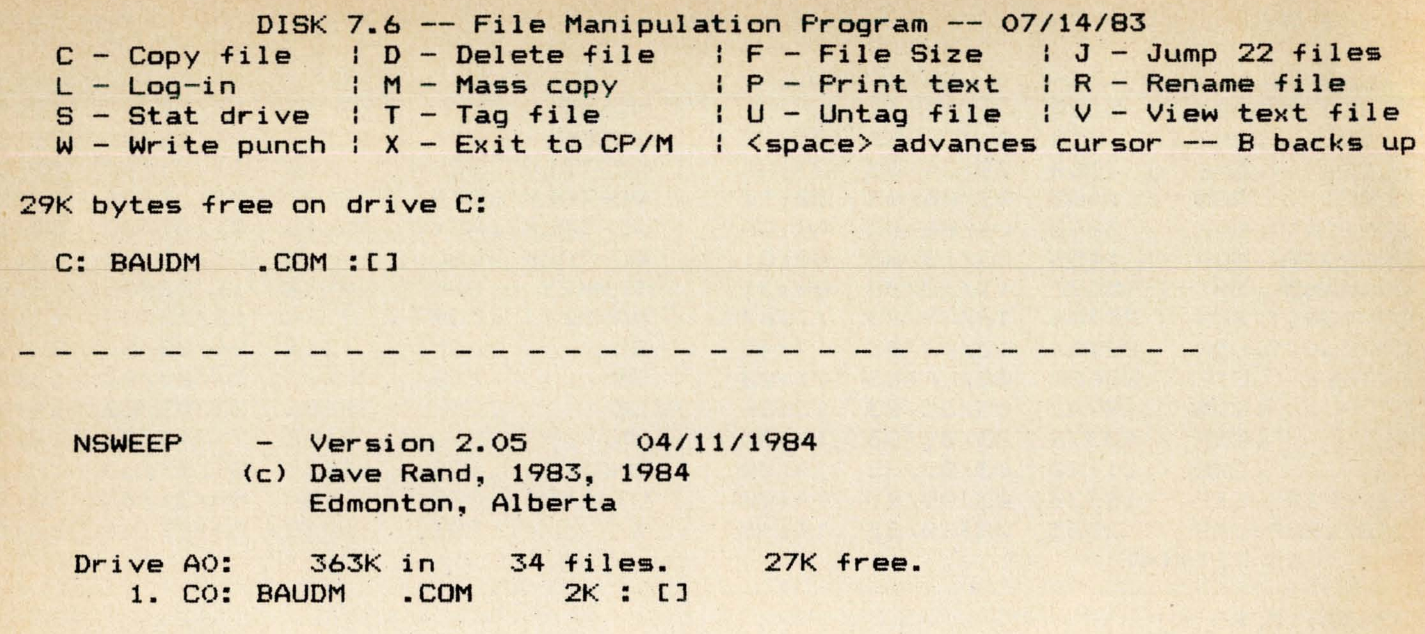
I'm convinced that D.COM and its forms are far superior to CP/M's DIR and STAT. While STAT may be needed for tasks like setting attributes, the average user should quickly gain from D.COM.

DISK.COM and NSWP.COM

DISK.COM and NSWP.COM are

DISK.COM		NSWP.COM	
		A	Retag files
B	Backs up	B	Back one file
C	Copy file	C	Copy file
D	Delete file	D	Delete file
		E	Erase T/U files
F	File size	F	Find files
J	Jump 22 files		
L	Log-in	L	Log new disk/user
M	Mass copy	M	Mass file copy
P	Print text	P	Print file
		Q	Squeeze/Unsqueeze tagged files
R	Rename file	R	Rename file(s)
S	Stat drive	S	Check remaining space
T	Tag file	T	Tag file for transfer
U	Untag file	U	Untag file
V	View text file	V	View file
W	Write punch	W	Wildcard Tag of files
X	Exit to CP/M	X	Exit to CP/M
		Y	Set file status
		?	Display this help
space	Advance cursor	sp, cr	Forward one file

Table 1. *The options for DISK.COM and NSWP.COM.*



```
            DISK 7.6 -- File Manipulation Program -- 07/14/83
   C - Copy file    | D - Delete file     | F - File Size     | J - Jump 22 files
   L - Log-in       | M - Mass copy       | P - Print text    | R - Rename file
   S - Stat drive   | T - Tag file        | U - Untag file    | V - View text file
   W - Write punch  | X - Exit to CP/M    | <space> advances cursor -- B backs up

 29K bytes free on drive C:

   C: BAUDM   .COM :[]



 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -


    NSWEEP     -  Version 2.05       04/11/1984
              (c) Dave Rand, 1983, 1984
                  Edmonton, Alberta

    Drive A0:     363K in    34 files.      27K free.
       1. C0: BAUDM    .COM      2K : []
```

Figure 9. *The start-up screens for DISK.COM and NSWP.COM.*

two public domain utility programs that perform almost every file manipulation commonly done. Although you might expect that DISK.COM and NSWP.COM are rival programs (they share many of the same operations), each has something special to offer. If they were commercial programs they *would* be rivals; since they are free public domain programs, both are worth having.

Between the two programs 19 different options are offered. These options replace eight CP/M commands — PIP, ERA, DIR, USER, TYPE, Control-P, REN and STAT. They present these operations in menus and they use prompts. And more, they offer features which make dealing with files easier and less painful.

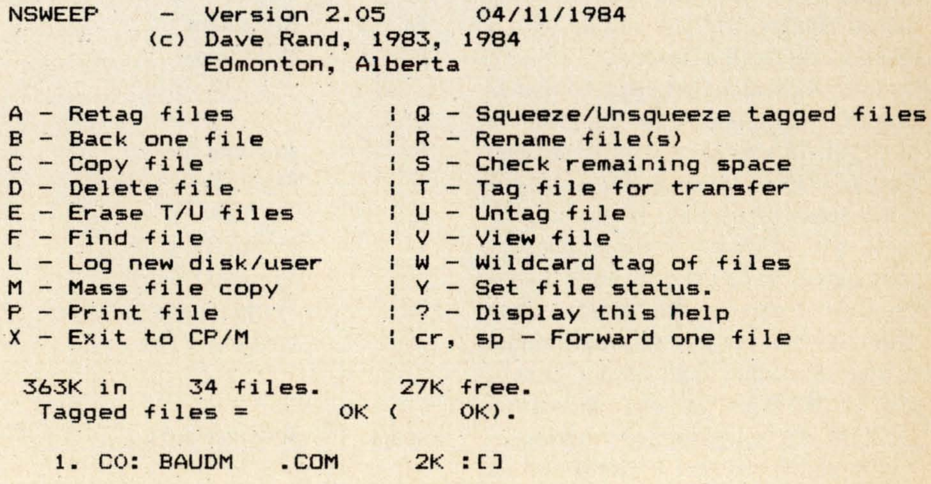
DISK.COM and NSWP.COM features can be grouped into six categories: cursor movement, tagging files, copying files, deleting files, text files, and everything else. These categories mimic much of the work we do with and to files.

The options for DISK.COM and NSWP.COM are presented in Table 1. Compare each option in the table as we go through them. Think about which program is best for your work (or if you want both).

What You See First

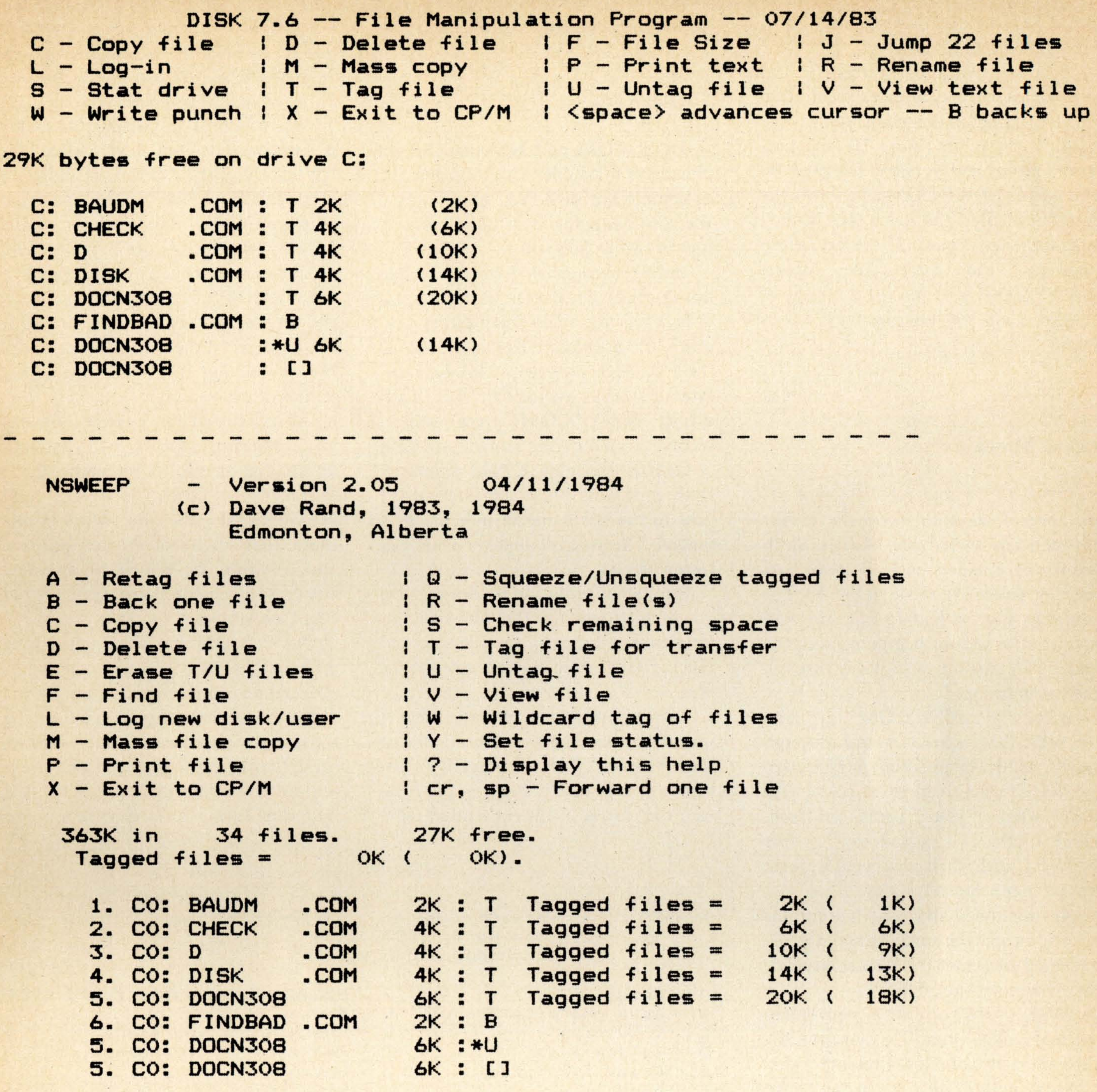
Figure 9 shows the initial screen information for both DISK.COM and NSWP.COM. The primary difference between them is that DISK.COM displays a menu immediately while NSWP.COM does not. NSWP.COM's menu displays after you enter a question mark. (You need not press Return after entering the letter for a menu option, in either NSWP.COM or DISK.COM.)

DISK.COM displays the program's name, its version number and the date the program was last changed. NSWP.COM shows the same information, but also includes the author's name and copyright notice. Authors of public domain programs like NSWP.COM often copyright their programs. Putting copyright notices in the software is a way to deter some unscrupulous software and hardware vendors and individuals from making a profit on other people's work. The program is in the public domain, but it is



```
NSWEEP     -  Version 2.05       04/11/1984
          (c) Dave Rand, 1983, 1984
              Edmonton, Alberta

A - Retag files            | Q - Squeeze/Unsqueeze tagged files
B - Back one file          | R - Rename file(s)
C - Copy file              | S - Check remaining space
D - Delete file            | T - Tag file for transfer
E - Erase T/U files        | U - Untag file
F - Find file              | V - View file
L - Log new disk/user      | W - Wildcard tag of files
M - Mass file copy         | Y - Set file status.
P - Print file             | ? - Display this help
X - Exit to CP/M           | cr, sp - Forward one file

 363K in    34 files.     27K free.
  Tagged files =       0K (     0K).

   1. C0: BAUDM   .COM     2K :[]
```

Figure 10. *The menu display for NSWP.COM.*



```
             DISK 7.6 -- File Manipulation Program -- 07/14/83
  C - Copy file    | D - Delete file    | F - File Size    | J - Jump 22 files
  L - Log-in       | M - Mass copy      | P - Print text   | R - Rename file
  S - Stat drive   | T - Tag file       | U - Untag file   | V - View text file
  W - Write punch  | X - Exit to CP/M   | <space> advances cursor -- B backs up

29K bytes free on drive C:

  C: BAUDM   .COM : T 2K      (2K)
  C: CHECK   .COM : T 4K      (6K)
  C: D       .COM : T 4K     (10K)
  C: DISK    .COM : T 4K     (14K)
  C: DOCN308      : T 6K     (20K)
  C: FINDBAD .COM : B
  C: DOCN308      :*U 6K     (14K)
  C: DOCN308      : []
```

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

   NSWEEP    -  Version 2.05      04/11/1984
           (c) Dave Rand, 1983, 1984
               Edmonton, Alberta

   A - Retag files            | Q - Squeeze/Unsqueeze tagged files
   B - Back one file          | R - Rename file(s)
   C - Copy file              | S - Check remaining space
   D - Delete file            | T - Tag file for transfer
   E - Erase T/U files        | U - Untag file
   F - Find file              | V - View file
   L - Log new disk/user      | W - Wildcard tag of files
   M - Mass file copy         | Y - Set file status.
   P - Print file             | ? - Display this help
   X - Exit to CP/M           | cr, sp - Forward one file

    363K in    34 files.      27K free.
     Tagged files =       0K (     0K).

       1. C0: BAUDM   .COM    2K : T  Tagged files =    2K (    1K)
       2. C0: CHECK   .COM    4K : T  Tagged files =    6K (    6K)
       3. C0: D       .COM    4K : T  Tagged files =   10K (    9K)
       4. C0: DISK    .COM    4K : T  Tagged files =   14K (   13K)
       5. C0: DOCN308         6K : T  Tagged files =   20K (   18K)
       6. C0: FINDBAD .COM    2K : B
       5. C0: DOCN308         6K :*U
       5. C0: DOCN308         6K : []
```

Figure 11. *Both programs let you tag and untag files for group operations.*

also owned by the true author, which means it cannot be sold without the author's permission — it can only be given away for free.

Next, DISK.COM displays its menu and tells you how much space (in bytes) is free on the current drive. If you change to another drive, DISK.COM displays the same information for the new drive.

DISK.COM then shows you the first file on the current user and drive in ASCII sequence. The drive letter and user number is shown to remind you of the user number and drive, with the exception of user 0 of the drive. It displays without the user number shown.

NSWP.COM displays much the same information, but numbers each file and shows you how many files are on the current user number and drive. It also shows how much space they occupy on the disk. Unlike DISK.COM, NSWP.COM does not suppress the "0" when files are displayed for user 0. Also unlike DISK.COM, NSWP.COM also tells you the size of the file without asking.

When first starting up NSWP.COM, if you enter a question mark to bring up the menu, you get the display shown in figure 10: the program name and version, followed by the menu. NSWP.COM also shows the number of files, the space they occupy and total free space. However, before showing the first file name, NSWP.COM tells you the total size of tagged files. We describe tagged files later.

Cursor Movement

DISK.COM and NSWP.COM offer several ways to move the cursor through the sorted list of files. Both programs make it easy to move forward or backward through the list. For example, if a list had 45 files in it and you want to see one beginning with the letter 'T', moving backward through the list is faster.

Both DISK.COM and NSWP.COM move forward through the list of files by pressing the spacebar. NSWP.COM also advances to the next file when you press Return. DISK.COM advances like NSWP.COM when you press Return, but its menu doesn't say so.

As mentioned above, both programs let you move backward through the list of files. Just press `B` (no Return needed) and you move back one file.

DISK.COM offers a convenient method called "jump" to move rapidly through a list of files. Pressing `J` displays the next 22 files. If you don't have 22 files left in the list, it starts the list over again. This command is handy when you're browsing through a list trying to remember the name of a file.

NSWP.COM has a similar option called "find" which moves you rapidly through the list. Pressing `F` prompts you for the file name you want to locate. You may use wildcards or just the first few letters of the file name if you want. When you know all or part of the file name, this is a quick way to move to a file.

Which is better is a toss up. I find both useful in different situations.

Tagging Files

DISK.COM and NSWP.COM let you tag or mark files for group copying or some other task. You may untag a file also. Tagging is a quick way to identify what files you want to copy.

Both programs use `T` to tag and `U` to untag files, as shown in figure 11. When you tag a file using either program, the program tells you the size of the file. Both programs also accumulate file sizes as you tag each file. In addition, NSWP.COM gives you a second size indicating an accumulation of the true size before CP/M rounding. Both programs place an asterisk (*) next to the file name to indicate it is tagged. The asterisk appears when you review the list.

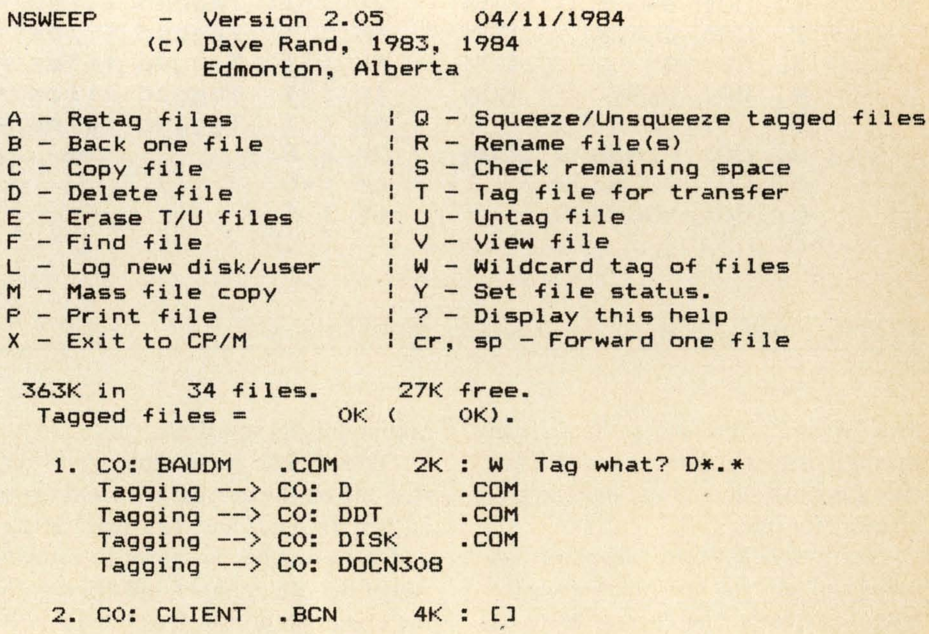
Similar information is shown when you press `U` to untag a file. The accumulated size reduces by the file size and the tag (indicated by the asterisk) is removed.

The tagging and untagging of files is useful for backing up several related files. You don't have to enter PIP or parameters repeatedly. Just tag the files, use the Mass Copy operation (explained below), then untag them. If you're finished with DISK.COM or NSWP.COM, you can untag the files by just leaving the program.

NSWP.COM in comparison offers a fast way to tag files called the "Wildcard Tag," shown as option `W` (figure 12). NSWP.COM asks you what you want to tag, and can use the CP/M "wildcards" for file matching — question mark (?) and asterisk (*). NSWP.COM shows all matches as they are tagged. This feature dramatically speeds up the process when copying or deleting files with similar file names or extensions.

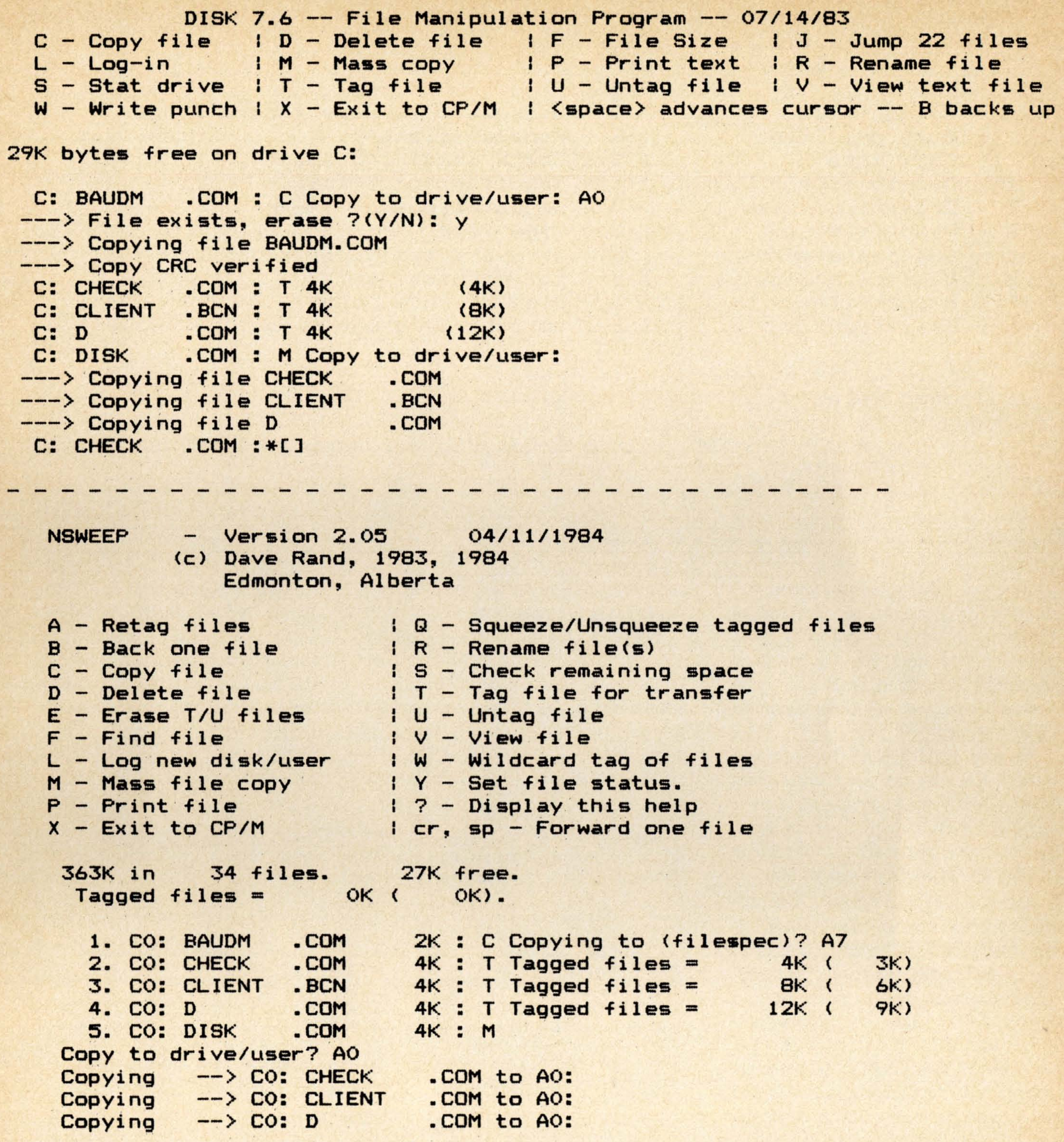
As mentioned above, after a file has been tagged, it appears in the list with an asterisk next to its file name. Once tagged, DISK.COM and NSWP.COM lets you Mass Copy these files to another user number and/or drive. NSWP.COM also lets you erase or squeeze/unsqueeze tagged files (see below).

NSWP.COM handles tagged files differently after a task like copying or squeezing is performed. It changes the asterisk to a pound sign (#), indicating the files have been used. You may retag these files by using the `A` option. These features are highly desirable in the late hours or during busy days when you



```
NSWEEP     -  Version 2.05       04/11/1984
         (c) Dave Rand, 1983, 1984
             Edmonton, Alberta

A - Retag files             ¦ Q - Squeeze/Unsqueeze tagged files
B - Back one file           ¦ R - Rename file(s)
C - Copy file               ¦ S - Check remaining space
D - Delete file             ¦ T - Tag file for transfer
E - Erase T/U files         ¦ U - Untag file
F - Find file               ¦ V - View file
L - Log new disk/user       ¦ W - Wildcard tag of files
M - Mass file copy          ¦ Y - Set file status.
P - Print file              ¦ ? - Display this help
X - Exit to CP/M            ¦ cr, sp - Forward one file

 363K in    34 files.     27K free.
  Tagged files =      0K (    0K).

   1. C0: BAUDM   .COM     2K : W  Tag what? D*.*
      Tagging --> C0: D       .COM
      Tagging --> C0: DDT     .COM
      Tagging --> C0: DISK    .COM
      Tagging --> C0: DOCN308

   2. C0: CLIENT  .BCN     4K : []
```

Figure 12. *NSWP.COM has a faster "wildcard tag" feature.*



```
             DISK 7.6 -- File Manipulation Program -- 07/14/83
  C - Copy file     | D - Delete file     | F - File Size     | J - Jump 22 files
  L - Log-in        | M - Mass copy       | P - Print text    | R - Rename file
  S - Stat drive    | T - Tag file        | U - Untag file    | V - View text file
  W - Write punch   | X - Exit to CP/M    | <space> advances cursor -- B backs up

29K bytes free on drive C:

  C: BAUDM   .COM : C Copy to drive/user: A0
 ---> File exists, erase ?(Y/N): y
 ---> Copying file BAUDM.COM
 ---> Copy CRC verified
  C: CHECK   .COM : T 4K          (4K)
  C: CLIENT  .BCN : T 4K          (8K)
  C: D       .COM : T 4K         (12K)
  C: DISK    .COM : M Copy to drive/user:
 ---> Copying file CHECK    .COM
 ---> Copying file CLIENT   .BCN
 ---> Copying file D        .COM
  C: CHECK   .COM :*[]

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
   NSWEEP    -  Version 2.05      04/11/1984
           (c) Dave Rand, 1983, 1984
               Edmonton, Alberta

   A - Retag files               | Q - Squeeze/Unsqueeze tagged files
   B - Back one file             | R - Rename file(s)
   C - Copy file                 | S - Check remaining space
   D - Delete file               | T - Tag file for transfer
   E - Erase T/U files           | U - Untag file
   F - Find file                 | V - View file
   L - Log new disk/user         | W - Wildcard tag of files
   M - Mass file copy            | Y - Set file status.
   P - Print file                | ? - Display this help
   X - Exit to CP/M              | cr, sp - Forward one file

    363K in    34 files.     27K free.
     Tagged files =      0K (    0K).

       1. C0: BAUDM   .COM     2K : C Copying to (filespec)? A7
       2. C0: CHECK   .COM     4K : T Tagged files =       4K (   3K)
       3. C0: CLIENT  .BCN     4K : T Tagged files =       8K (   6K)
       4. C0: D       .COM     4K : T Tagged files =      12K (   9K)
       5. C0: DISK    .COM     4K : M
    Copy to drive/user? A0
    Copying   --> C0: CHECK    .COM to A0:
    Copying   --> C0: CLIENT   .COM to A0:
    Copying   --> C0: D        .COM to A0:
```

Figure 13. *Both DISK.COM and NSWP.COM let you copy files to any drive and any user area.*

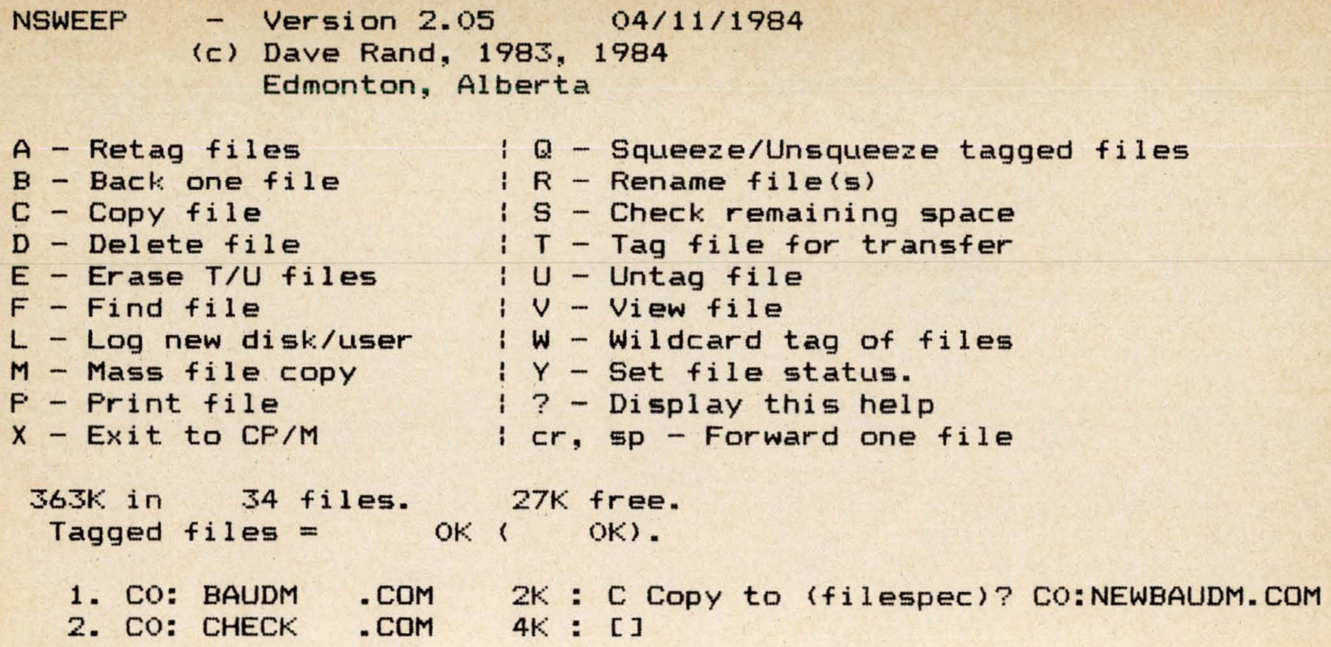
have trouble remembering what happened 30 seconds before.

Copying Files

DISK.COM and NSWP.COM both offer options to copy one or more files. Figure 13 shows how to copy a file from one user number and drive to another using either program.

Copy (`C`) in both programs transfers the current file in the list to the drive and user number desired. However, you may *not* transfer a file to the same drive and user (see NSWP.COM exception below). Both programs ask what drive and user number you want the file copied to.

DISK.COM reminds you what file it is copying and tells you the transfer is



```
NSWEEP     -  Version 2.05      04/11/1984
         (c) Dave Rand, 1983, 1984
             Edmonton, Alberta

A - Retag files             | Q - Squeeze/Unsqueeze tagged files
B - Back one file           | R - Rename file(s)
C - Copy file               | S - Check remaining space
D - Delete file             | T - Tag file for transfer
E - Erase T/U files         | U - Untag file
F - Find file               | V - View file
L - Log new disk/user       | W - Wildcard tag of files
M - Mass file copy          | Y - Set file status.
P - Print file              | ? - Display this help
X - Exit to CP/M            | cr, sp - Forward one file

 363K in    34 files.      27K free.
  Tagged files =       OK (     OK).

    1. CO: BAUDM    .COM    2K : C Copy to (filespec)? CO:NEWBAUDM.COM
    2. CO: CHECK    .COM    4K : []
```

Figure 14. *NSWP.COM lets you change the name of the file during a copy operation.*

`CRC Verified`. CRC stands for Cyclic Redundancy Check. It is a method of checking accurate transfer of files. NSWP.COM does not say anything about CRC checking, but it may nonetheless do this common checking method as it copies.

In addition, DISK.COM warns you when a file with the same name exists on the disk you're copying to. This is very useful after many hours of work. A careless slip of a finger in NSWP.COM could wipe it all out.

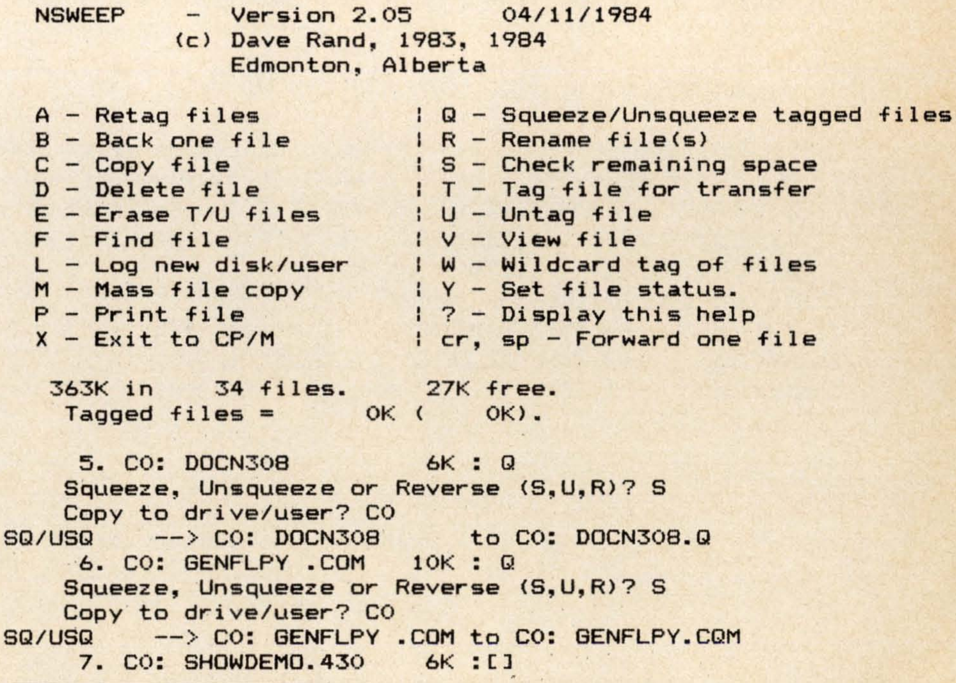
DISK.COM and NSWP.COM differ in a significant way, as shown in figure 14. NSWP.COM lets you change the name of the file during transfer, and if you do change the name, you can copy it to the same user number and drive. BAUDM.COM in figure 14 is an example of simultaneously transferring a file to the same user number and drive, and renaming the file. If you need to keep separate versions of documents or programs, this feature is particularly useful.

Both programs let you copy all tagged files. The files can be "mass-copied" to another user number, or another user number and drive. Figure 13 shows examples.

When "mass copying," *neither DISK.COM nor NSWP.COM check the receiving disk to determine if a file with the same name exists.* This is a weakness in both programs.

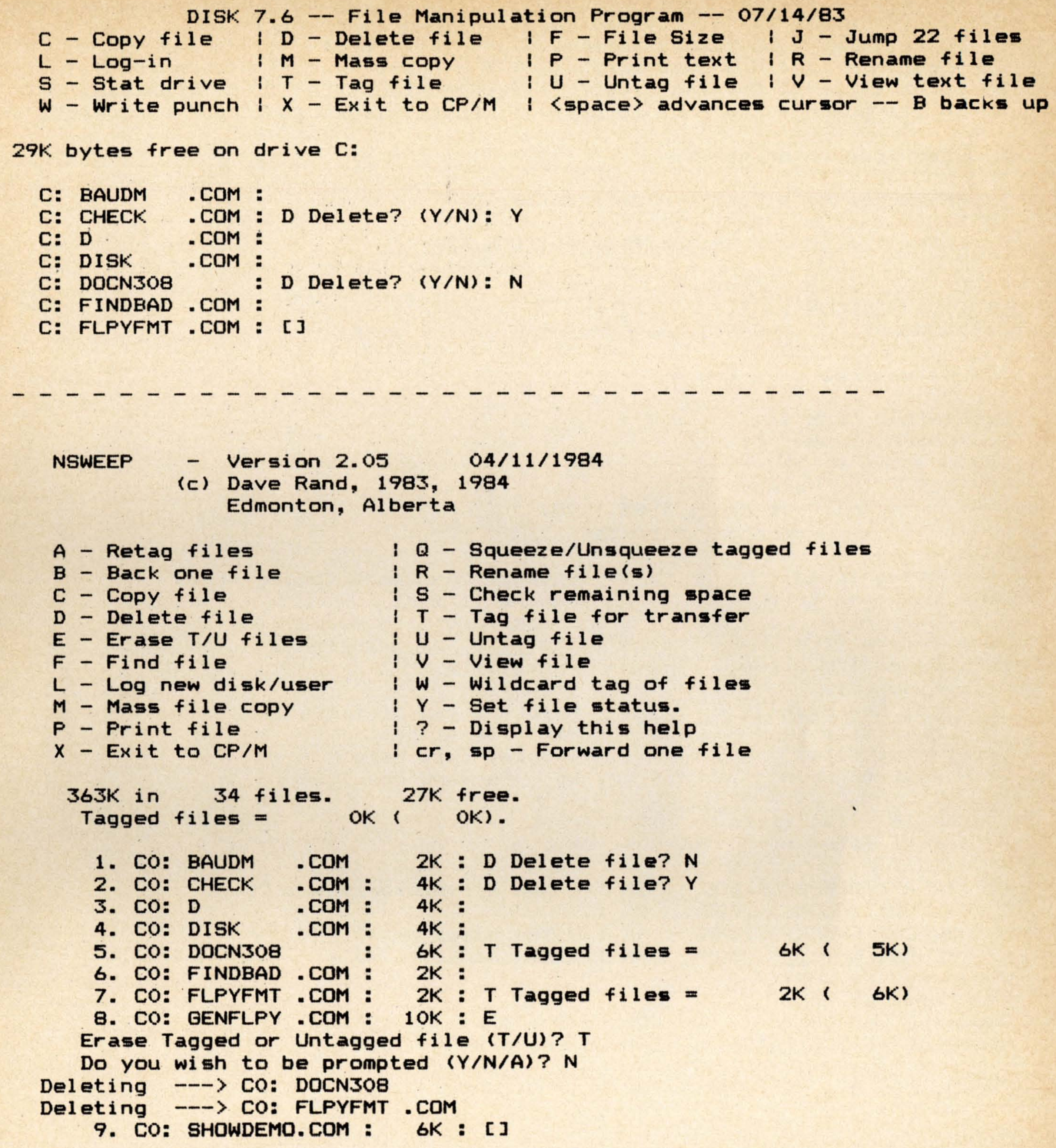
NSWP.COM offers a unique option not offered by CP/M or by DISK.COM. The Squeeze/Unsqueeze option (see figure 15) compresses files so that they take up less space. It doesn't change the original file, but creates a new one with a similar file name and extension. In my testing this feature reduced the size of various files by up to 40%. Obviously, text files like those created with a word processor are the most dramatic, yet both compiled and interpreted programs also are reduced.

Squeezing is useful when backing up word processing and source pro-



```
    NSWEEP     -  Version 2.05      04/11/1984
             (c) Dave Rand, 1983, 1984
                 Edmonton, Alberta

    A - Retag files             | Q - Squeeze/Unsqueeze tagged files
    B - Back one file           | R - Rename file(s)
    C - Copy file               | S - Check remaining space
    D - Delete file             | T - Tag file for transfer
    E - Erase T/U files         | U - Untag file
    F - Find file               | V - View file
    L - Log new disk/user       | W - Wildcard tag of files
    M - Mass file copy          | Y - Set file status.
    P - Print file              | ? - Display this help
    X - Exit to CP/M            | cr, sp - Forward one file

     363K in    34 files.      27K free.
      Tagged files =       OK (     OK).

        5. CO: DOCN308         6K : Q
      Squeeze, Unsqueeze or Reverse (S,U,R)? S
      Copy to drive/user? CO
SQ/USQ     --> CO: DOCN308       to CO: DOCN308.Q
        6. CO: GENFLPY .COM    10K : Q
      Squeeze, Unsqueeze or Reverse (S,U,R)? S
      Copy to drive/user? CO
SQ/USQ     --> CO: GENFLPY .COM to CO: GENFLPY.CQM
        7. CO: SHOWDEMO.430     6K :[]
```

Figure 15. *NSWP.COM has the squeeze/unsqueeze option; squeezing a file compresses it to save disk space and transmission time.*



```
                DISK 7.6 -- File Manipulation Program -- 07/14/83
    C - Copy file    | D - Delete file    | F - File Size    | J - Jump 22 files
    L - Log-in       | M - Mass copy      | P - Print text   | R - Rename file
    S - Stat drive   | T - Tag file       | U - Untag file   | V - View text file
    W - Write punch  | X - Exit to CP/M   | <space> advances cursor -- B backs up

  29K bytes free on drive C:

    C: BAUDM   .COM :
    C: CHECK   .COM : D Delete? (Y/N): Y
    C: D       .COM :
    C: DISK    .COM :
    C: DOCN308      : D Delete? (Y/N): N
    C: FINDBAD .COM :
    C: FLPYFMT .COM : []

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

     NSWEEP    -  Version 2.05      04/11/1984
              (c) Dave Rand, 1983, 1984
                  Edmonton, Alberta

     A - Retag files            | Q - Squeeze/Unsqueeze tagged files
     B - Back one file          | R - Rename file(s)
     C - Copy file              | S - Check remaining space
     D - Delete file            | T - Tag file for transfer
     E - Erase T/U files        | U - Untag file
     F - Find file              | V - View file
     L - Log new disk/user      | W - Wildcard tag of files
     M - Mass file copy         | Y - Set file status.
     P - Print file             | ? - Display this help
     X - Exit to CP/M           | cr, sp - Forward one file

      363K in    34 files.     27K free.
       Tagged files =       0K (     0K).

        1. C0: BAUDM   .COM     2K : D Delete file? N
        2. C0: CHECK   .COM :   4K : D Delete file? Y
        3. C0: D       .COM :   4K :
        4. C0: DISK    .COM :   4K :
        5. C0: DOCN308      :   6K : T Tagged files =       6K (   5K)
        6. C0: FINDBAD .COM :   2K :
        7. C0: FLPYFMT .COM :   2K : T Tagged files =       2K (   6K)
        8. C0: GENFLPY .COM :  10K : E
       Erase Tagged or Untagged file (T/U)? T
       Do you wish to be prompted (Y/N/A)? N
    Deleting  ---> C0: DOCN308
    Deleting  ---> C0: FLPYFMT .COM
        9. C0: SHOWDEMO.COM :   6K : []
```

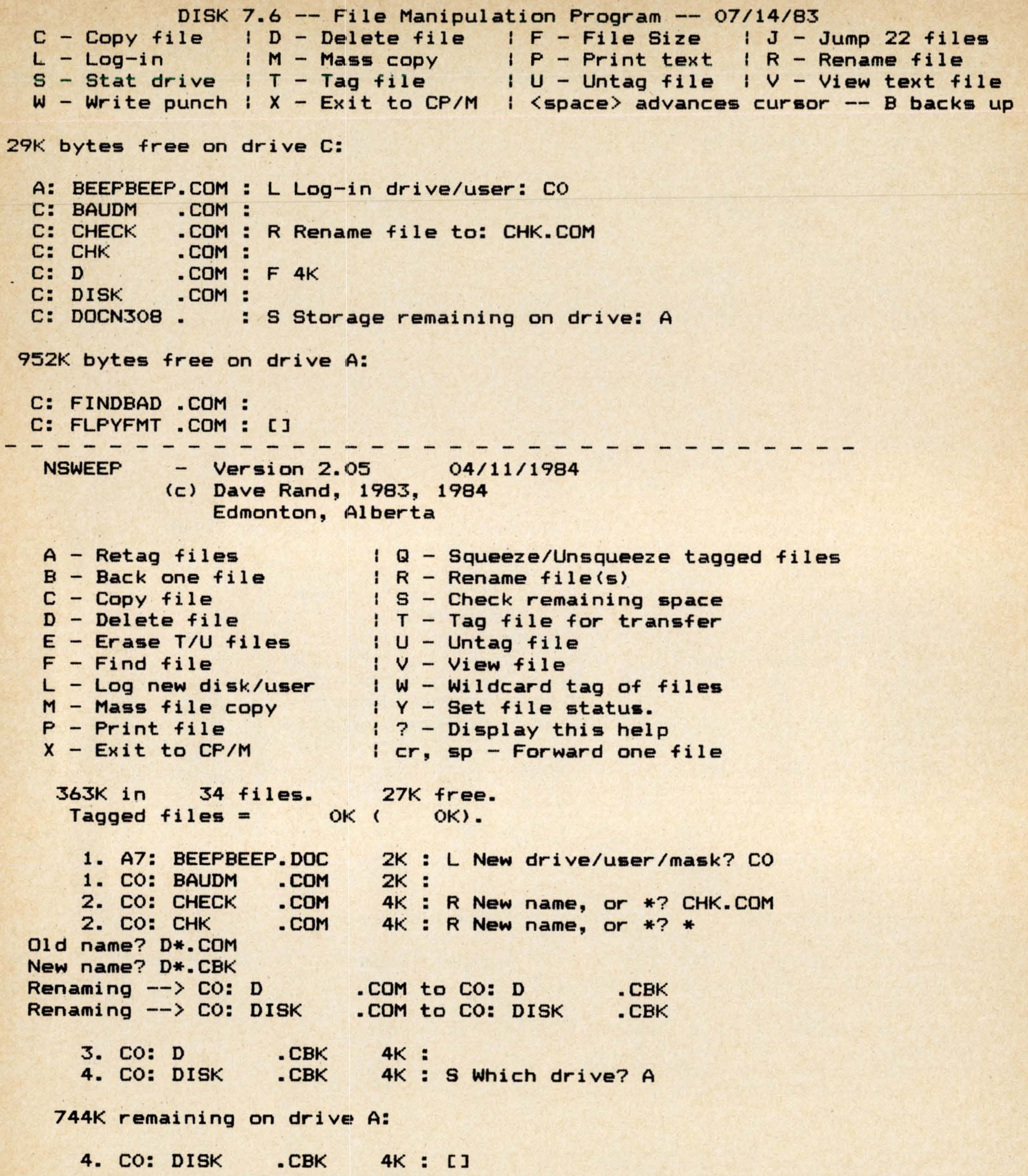
Figure 16. *Both programs offer an easy method of deleting files.*

grams to archive disks. The reduced space a compressed file uses saves many dollars in disks.

This feature is also useful when transferring files between computers over phone lines. The reduced size means less time is needed to transfer the file, and therefore, less phone cost. Several public domain programs are available to squeeze and unsqueeze files, but this built-in feature makes them unnecessary.

Deleting Files

Both programs offer an easy method to delete files, as shown in figure 16. After choosing `D` to delete a file, answer



```
            DISK 7.6 -- File Manipulation Program -- 07/14/83
   C - Copy file    | D - Delete file    | F - File Size    | J - Jump 22 files
   L - Log-in       | M - Mass copy      | P - Print text   | R - Rename file
   S - Stat drive   | T - Tag file       | U - Untag file   | V - View text file
   W - Write punch  | X - Exit to CP/M   | <space> advances cursor -- B backs up

 29K bytes free on drive C:

   A: BEEPBEEP.COM : L Log-in drive/user: C0
   C: BAUDM   .COM :
   C: CHECK   .COM : R Rename file to: CHK.COM
   C: CHK     .COM :
   C: D       .COM : F 4K
   C: DISK    .COM :
   C: DOCN308 .    : S Storage remaining on drive: A

  952K bytes free on drive A:

   C: FINDBAD .COM :
   C: FLPYFMT .COM : []
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    NSWEEP     -  Version 2.05      04/11/1984
           (c) Dave Rand, 1983, 1984
               Edmonton, Alberta

    A - Retag files             | Q - Squeeze/Unsqueeze tagged files
    B - Back one file           | R - Rename file(s)
    C - Copy file               | S - Check remaining space
    D - Delete file             | T - Tag file for transfer
    E - Erase T/U files         | U - Untag file
    F - Find file               | V - View file
    L - Log new disk/user       | W - Wildcard tag of files
    M - Mass file copy          | Y - Set file status.
    P - Print file              | ? - Display this help
    X - Exit to CP/M            | cr, sp - Forward one file

     363K in    34 files.    27K free.
      Tagged files =      OK (    OK).

       1. A7: BEEPBEEP.DOC    2K : L New drive/user/mask? C0
       1. C0: BAUDM   .COM    2K :
       2. C0: CHECK   .COM    4K : R New name, or *? CHK.COM
       2. C0: CHK     .COM    4K : R New name, or *? *
   Old name? D*.COM
   New name? D*.CBK
   Renaming --> C0: D        .COM to C0: D       .CBK
   Renaming --> C0: DISK     .COM to C0: DISK    .CBK

       3. C0: D       .CBK    4K :
       4. C0: DISK    .CBK    4K : S Which drive? A

     744K remaining on drive A:

       4. C0: DISK    .CBK    4K : []
```

Figure 17. *Other commands featured in DISK.COM and NSWP.COM include renaming, changing drives, and displaying file size.*

`Y` (yes) to delete the current file, or `N` (no) if you change your mind. The only difference in the way the programs delete files is DISK.COM shows your options — (Y/N).

NSWP.COM offers the option to erase a group of files. You may erase all tagged or all untagged files. With the `prompt (Y/N/A)` option, you can selectively delete tagged or untagged files with a Y (yes answer). The A answer lets you abort this option.

Be careful with both types of deleting, because unless you have a program to "undelete" a file or an accurate backup copy, you'll never see the file again.

Viewing/Printing Text Files

DISK.COM and NSWP.COM offer two ways to look at text files created by most common word processors or editors. They are View and Print. As the names indicate, View (V) displays the text on the screen so you can view it, and Print (P) sends it to your printer.

The View (V) option has two minor differences in the way they work in DISK.COM and NSWP.COM. DISK.COM uses the spacebar to advance the text one line while NSWP.COM uses the letter L. The other difference is that DISK.COM says to press any other key to display the next 22 lines of text. NSWP.COM says to press Return, but any key other than L does the same thing.

In brief, in either program press Control-X to stop viewing the text, or any key to see the next 22 lines. Press the spacebar in DISK.COM or L in NSWP.COM to see the next line of text.

The Print (P) option in both programs work the same. They send the text to your printer. If you have two printers connected to your computer, you'll first have to tell CP/M which printer to use. DISK.COM and NSWP.COM just tell CP/M to print the text; CP/M has to know where.

Other Options

DISK.COM and NSWP.COM offer several other options. These are shown in figure 17.

Both DISK.COM and NSWP.COM let you change between user numbers and/or drives. The L option in either program makes it very easy to search your hard disk for files and do several task involving different user numbers and drives. The programs would be dramatically less beneficial without this option.

Renaming files is similar in both programs. You simply choose R, then tell it the new name. NSWP.COM has an added feature, though. It lets you rename files using wildcards. For example, all the files which make up this article have the ".UG" extension. I may want to rename them as I develop new versions. You can probably think of several uses yourself.

DISK.COM has an option which tells you the size of the current file. When determining which files will fit on a back up disk, the file's size is critical. NSWP.COM provides this information automatically.

Both programs have an option to determine how much space is left on a disk, whether a floppy or hard disk. This is also important when making back up copies.

Finally, if you review what we've gone over you'll notice I didn't cover several options. The Write Punch option in DISK.COM was not mentioned because the days of punched paper tape as a storage medium are essentially over. The Set Status option in NSWP.COM doesn't appear to work. At least I can't make it work.

DISK or NSWP: Which is better?

This question is like being asked whether you like vanilla or chocolate ice cream, when you actually prefer fudge ripple. I've been using DISK.COM since November 1983. It does things very well but has a few quirks. NSWP.COM has more features and some faster operating features like wild-card tagging.

Do I prefer one over the other? I want both, and more! While I'm not a programmer, there are many who are and who enjoy creating innovative utility programs (among others). And if the history of public domain programs is any indication, someone will create a good cross between them. Then I'll have fudge ripple.

ACTIVE TRACE

"A marvelous Basic programming aid . . .

It's just amazing to watch a program you wrote run under Scope, and debugging becomes if not trivial, then at least doable."

Thomas Bonoma, *Microcomputing*
Dec. '83, p 22

"Extremely useful program . . . Anyone doing much programming in Basic should appreciate Active Trace a lot."

Jerry Pournelle, *Byte Magazine*
April '83, p 234

Spaghetti code is what many "experts" call a beginner's Basic program which is all tangled up and difficult to follow. The **Active Trace** package will help you learn how to avoid the pitfalls of structureless programs. And if you already have a program which is too confusing to follow, or has an error which is hiding, relax. **Active Trace** doesn't get confused. **Active Trace** will lead you through your program letting you know variable values (all variables or just those you specify) as they change. In a form a novice can understand, your program's internal activity is presented on your screen, or printer, or it can be saved on disk.

For more advanced programmers, the disk file of your programs activity can be used with your word processor to automatically find the source of an error and display the circumstances surrounding its occurrence.

Ready to Order?
Just have a Question?
Contact your dealer or call
Toll Free: 800-358-9120(US)
800-862-4948(CA)

Active Trace **$79.95**
Complete package includes Scope, XREF mapping and documentation

Scope Separately **$49.95**
Only recommended for those who already own professional mapping (XREF) programs

Active Trace is available for most **MS-DOS** and **CPM 2.2** systems and supports the special features of Brand specific versions of Microsoft Basic such as **Basica** on the **IBM-PC**.

AWARECO
Active Software

P.O. Box 695 Gualala, CA 95445
(707) 884-4019

Active trace, Active software, AWARECO and Scope are trademarks of A Ware Company—CPM is a trademark of Digital Research—MS-DOS and Microsoft are trademarks of Microsoft Corporation—IBM-PC is a trademark of IBM Corp.

Circle 1 on the reader service card.

CP/M-80 •
CP/M-86 •
PC-DOS •
MS-DOS •

Turbo Pascal
(Part 1)

This is part one of a tutorial on Pascal programming, featuring the standard Pascal language and the popular Turbo Pascal compiler.

We have found Turbo Pascal to be an excellent programming tool for our applications. Although we consider ourselves "amateur" programmers, we think the package will also be valuable to professional programmers.

What we like most about Turbo Pascal is that it makes the Pascal language accessible to anyone who can afford the $50 price tag. Unlike other inexpensive programming languages (including JRT Pascal), the Turbo Pascal package can be used by anyone with minimal knowledge of WordStar editing commands.

Our first and lasting impression is that this is a tool for anyone using a personal computer. We did not think that only programmers would understand it — as we usually think about most programming language packages. Turbo Pascal was a "commoner's" Pascal that not only worked, it worked *fast.*

Programs compiled by the Turbo Pascal compiler are much faster than similarly-complex BASIC programs. You can create, compile and run a program nearly as fast as using MBASIC to perform the same task interactively. Turbo Pascal contains a text editing function that combines WordStar's most useful commands, then gives you the ability to compile and execute the program on the spot. If there are errors during the compile, Turbo Pascal puts you back in the edit mode on the line causing the error. You can make *and fix* programs faster than in most other languages we've used.

This tutorial introduces the Turbo Pascal compiler and editor, and the Pascal language, in a very simple application: moving data from one record format to another. Most Pascal tutorials show off the language's inherent features that let you write very sophisticated programs. We have instead decided to introduce Pascal in the context of a simple problem to solve — a problem that shows the most typical of Pascal operations: moving data from one file to another. The next installment will go deeper into the language's features.

Running Turbo

After installing Turbo Pascal (run TINST and pick your terminal or computer), you run it by typing the name `TURBO` ↵. This produces the following:

```
TURBO Pascal Version 1.00
Copyright (C) 1983 by BORLAND International

Televideo 950

Include error messages (Y/N)?
```

Type `Y` to include error messages because they help you find bugs in your program. This menu should appear:

```
Logged drive: A

Work file:
Main file:
```

Typefaces

We use `This Typeface` to show commands and expressions you type at your keyboard, and `this typeface` to show displayed messages. In this article we also use `UPPER` case letters for Pascal reserved words.

We also use the symbol ↵ to represent the standard Return key (CR or ENTER on some keyboards), and the caret symbol ^ to represent the Control key (CNTL, CTRL or ALT on some keyboards).

by Tony Bove & Cheryl Rhodes

```
Edit        Compile     Run     Save

eXecute     Dir         Quit    compiler Options

Text:     0 bytes (7714-7714)
Free: 25841 bytes (7715-DC06)
```

You start with the `E` option to edit a new work file. Turbo Pascal asks for a filename, and puts you into edit mode (a blank screen). You can now type your program statements and use the familiar WordStar commands to edit them.

You have several advantages in Turbo Pascal's edit mode over WordStar, starting with the ability to "undo" a line. If you accidently delete a line with Control-`Y`, or change it without leaving it, you can restore it with the Control-`Q` Control-`L` combination *as long as the cursor has not left the line.*

Another advantage is the ability to move the cursor up and down lines without the cursor jumping to the last word on each line. It appears that the screen is full of spaces, but this is not the case. The ease of moving the cursor makes it easier to line up `Begin` and `End` statements.

The Outline

A task is usually easier to understand when you draw an outline of it. To show how Turbo Pascal can solve simple problems quickly, we present a simple problem we were vexed with: how to convert a text file with names and addresses in a particularly odd record format to another record format that is compatible with dBASE II and MailMerge (two popular data base management programs).

It doesn't actually matter what the details are about the two formats. You can follow along and draw the following outline that solves the problem:

```
Begin
        Set up in-file (file with data to convert);
        Set up out-file (new file with converted data);
        while there is data (not at the end of file), do
           begin
              read five lines of the text file;
              form them into one line with delimiters;
              write the new line to the new file;
           end;
end.
```

With this outline you have already started using Pascal statements. "Begin" and "end" are the boundaries of a Pascal procedure or routine, and you use them twice: first to define the entire procedure, and again inside the procedure to define a routine. You also used several other Pascal words: "while", "do", "read" and "write".

With the above outline on your screen, type Control-`KD` to return to the menu. Unlike WordStar's `^KD` that saves the file before returning to the menu, this `^KD` returns you to the menu. To save the outline, type the `S` option, which automatically saves your work file by the name you used. If you want to save the outline by another name, you can change the work file name with the `W` option.

The Program

To return to edit mode, type `E`. You can now edit the outline into the Pascal program shown in figure 1. The data file's record format to be converted looks like this:

name1
name2
address
city
state
zip
subtype

Each record is formatted as shown above: seven separate lines each ending with a Return key (you can set up a test data base using a word processing program or using Turbo Pascal's edit mode).

The program we wrote converts the data into this new format:

name1,name2,address,city,state,zip,subtype

Figure 1 shows the Pascal program that performs the conversion. Figure 2a shows the input file to be converted, and 2b shows the resulting output file with the converted data.

All Pascal programs start with the word `PROGRAM` and end with the word `END`. Syntax (the spelling of words and the use of punctuation) is very important in Pascal. Each statement ends with a semicolon, which tells the compiler to interpret the words from the previous semicolon to the current one as one statement.

After `PROGRAM` we list any *constants* in the program. A constant is any number or string that does not change. Strings are enclosed in single quote marks ('); our only constant is the delimiter used in the new record format — a comma. Therefore, we define the constant `comma` as a string containing one comma.

The Turbo Pascal compiler parses `CONST comma = ',';` as one constant declaration statement, ignoring the spaces and indents. The compiler considers everything following it to be a constant declaration until it finds the word `VAR`. There happens to be only one constant declaration in this program.

Upon seeing `VAR`, the compiler interprets the next group of statements (up to `BEGIN`) as variable declarations. We have two types of variables in this program: `STRING` types each of a certain length, and a type called `TEXT`:

```
name1,name2,address,city,state,zip,subtype : STRING[30];
newrec : STRING[255];
infile, outfile : TEXT;
filename1, filename2 : STRING[14];
```

For each line of the data file shown in figure 2a, we have one string variable with a 30-character length. These variables, used throughout the program, are `name1`, `name2`, `address`, `city`, `state`, `zip` and `subtype`.

To represent the new format (shown in figure 2b) we have one 255-character string called `newrec`. This is a simple way to make sure we have a converted line with less than 256 characters. Pascal has more sophisticated methods for verifying the integrity of data that will be described later *(next installment)*.

Note that standard Pascal, as defined in 1971 by Professor Niklaus Wirth of the Technical Univ. of Zurich, did not include the variable type `STRING`, nor is the file type `TEXT` (described next) available. These are extensions to the language that make personal computing easier. The complete description of standard Pascal is the *Pascal User Manual and Report* (K. Jensen, N. Wirth), Springer-Verlag, 1974.

Why should you care if something is standard or not? There is no need to unless you are using the program in other environments besides the one you're using now.

For example, if you were writing a program to run under a completely different operating system, you'd need to use a different compiler (unless Turbo Pascal is available for that system). The other compiler may not recognize these extensions. But more often than not the other compiler either recognizes these extensions or uses similar ones to achieve the same result.

The Main Procedure

The main part of the program starts with the first `BEGIN` and proceeds to its corresponding `END.` (with a period). There can be more `BEGIN/END` blocks within the main one. There must be a corresponding `END;` (with a semicolon) for every `BEGIN` within the main block.

The first `WRITE` statement displays a message on the screen asking for a filename. The following `READLN` statement reads whatever you type until you type an "end of line" (a Return), and stores it as the variable `filename1`. The second `WRITE` statement asks for another file, and a `READLN` statement reads that filename. We are reading filenames into variables so that the program can convert a file with any name, and create a new one with any name.

After the `WRITE` and `READLN` statements are special *file procedures* that are pre-defined for the Turbo Pascal compiler. You call these built-in procedures by name and pass some useful *arguments* in parentheses to them. They perform a vital action that is not provided for in Wirth's standard Pascal:

```
ASSIGN(infile,filename1);
ASSIGN(outfile,filename2);
RESET(infile);
REWRITE(outfile);
```

The first two `ASSIGN` statements assign file variable names to the string variables used in the previous `READLN` statements. All subsequent file operations use these file variables.

The `RESET` statement moves an imaginary file pointer to the beginning of the file. By using this statement you make certain that the next file operation starts from the beginning of the file. `RESET` can only be used on a file that already exists.

The `REWRITE` statement creates a new file or deletes an existing one to start from scratch. It also moves the imaginary file pointer to the beginning of a file.

Repeating an Operation

We use the `WHILE condition DO` phrase to start a repetitive operation. There are two basic constructions:

```
WHILE condition DO statement

WHILE condition DO
      BEGIN
        statement;
        statement;
           .
           .
           .
      END;
```

The `condition` you specify can be a numeric expression like `counter<>0` (the counter is not zero) or a logical expression like `NOT EOF(infile)` (the file pointer is not at the end of the file `infile`).

We set up a block of statements to be executed over and over until the file pointer reaches the end of the file. At that point the `WHILE NOT EOF(infile) DO` phrase becomes inoperative, and the program continues after the `END;` statement that corresponds with the `BEGIN` that starts the `WHILE` block.

The seven `READLN` statements read each record of the text file `infile`. Following these statements, we use assignment statements to concatenate a comma to each string variable:

```
name1 := CONCAT(name1,comma);
name2 := CONCAT(name2,comma);
address := CONCAT(address,comma);
city := CONCAT(city,comma);
state := CONCAT(state,comma);
zip := CONCAT(zip,comma);
newrec := CONCAT(name1,name2,address,city,state,zip,subtype);
```

The symbol `:=` is best said as "becomes." The variable `name1` *becomes* the value expressed by `CONCAT(name1,comma)`. The built-in `CONCAT` function is similar to a BASIC built-in function — it returns a value, so it can be used on the right side of an assignment statement. The function concatenates two strings; the first in our case is the string holding data, and the second is the defined string constant `comma`.

We could have put a literal comma in each statement using the `CONCAT` function, as in `CONCAT(name1,',');`. However, we wrote this program to be as general as possible. If we needed a vertical bar (`¦`) as a delimiter rather than a comma, we could change the definition of `comma` to be `comma = '¦';`.

The last assignment statement concatenates all of the string variables together to form one string, `newrec`. This string is then displayed on the screen and subsequently written to the file `outfile` in these two statements:

```
WRITELN(newrec);
WRITELN(outfile,newrec);
```

The `WRITELN` statement automatically writes a Return at the end of the line.

The loop defined by the `WHILE NOT EOF(infile) DO BEGIN` and `END;` statements ends when the program reaches the end of the input file `infile`. The program then uses the built-in `CLOSE` procedure to close both files. Closing a file after using it is good practice because it assures that the last record written to the file was actually recorded on the disk (closing updates the directory entry properly).

Compiling and Running

Now comes the easy part. First type Control-`KD` to leave edit mode, then `S` to save the work file. On our computers (Alspa ACI, CompuPro 8:16, Osborne 1) Turbo Pascal saves the file much more quickly than WordStar.

To compile the program in memory, type `C`. If there are any errors, Turbo Pascal tells you in a brief error message what is wrong, and puts you back in edit mode *on the line of the error* so that you can fix it. If there are many errors, Turbo Pascal picks the first one to show you, in the hopes that in edit mode you will fix as many errors as you can plainly see.

Turbo Pascal is very good at finding and displaying errors during the compile process. The most common error is a misplaced or forgotten semicolon. Pascal statements can be on a separate line or on the same line, but they must be separated by a semicolon. If you forget one, the compiler rejects the program and returns you to edit mode to fix the line.

If the program passes the compiler, you'll see something like the following display:

```
Compiling
  36 lines
Code:   620 bytes (7BAD-7E19)
Free: 24210 bytes (7E19-DCAB)
Data:   859 bytes (DCAB-E006)
>
```

You can now run the program by typing `R`. If you make a mistake in typing data and the program makes an error, Turbo Pascal displays an appropriate error message and takes you into edit mode with the cursor on the line where the error occurred.

```
PROGRAM Convert;
CONST
      comma = ',';
VAR
      name1, name2, address, city, state, zip, subtype : STRING[30];
      newrec : STRING[255];
      infile, outfile : TEXT;
      filename1, filename2 : STRING[14];
BEGIN
      WRITE('Type name of file to convert: '); READLN(filename1);
      WRITE('Type name of file to create: '); READLN(filename2);
      ASSIGN(infile,filename1);
      ASSIGN(outfile,filename2);
      RESET(infile);
      REWRITE(outfile);
      WHILE NOT EOF(infile) DO
       BEGIN
         READLN(infile,name1);
         READLN(infile,name2);
         READLN(infile,address);
         READLN(infile,city);
         READLN(infile,state);
         READLN(infile,zip);
         READLN(infile,subtype);
         name1 := CONCAT(name1,comma);
         name2 := CONCAT(name2,comma);
         address := CONCAT(address,comma);
         city := CONCAT(city,comma);
         state := CONCAT(state,comma);
         zip := CONCAT(zip,comma);
         newrec := CONCAT(name1,name2,address,city,state,zip,subtype);
         WRITELN(newrec);
         WRITELN(outfile,newrec);
       END;
      CLOSE(infile); CLOSE(outfile);
END.
```

Figure 1. *A simple Turbo Pascal program to convert data from one record format (data file shown in figure 2a) to another record format (shown in 2b).*

Messages are more explicit during the compile process, because Turbo Pascal can guess what is wrong. In run-time, the error may not be as obvious. For example, if you typed the name of a file that did not exist as the input file, you would cause a non-Pascal error to occur:

```
I/O error 01, PC=00A9
Program aborted

Searching
  14 lines
Run-time error position found. Press <ESC>
```

When you press the ESC (Escape) key, Turbo Pascal puts you back into edit mode on the line where the error occurred.

We have not seen as good error recovery and setup for correction in any product. Turbo Pascal excels at making it easy for you to fix your source code, compile it, and run it again. You may have to fix another error, compile it, run it again, etc. until you get it right, but the task is much more pleasant than with other compilers and interpreters.

```
Tony Bove
User's Guide
2055 Woodside Rd. #180
Redwood City
CA
94061
C1
Cheryl Rhodes
TUG Inc.
P.O. Box 5245
Redwood City
CA
94063
C2
Don Larchmont
TUG Concert Promotion
P.O. Box 0001
Woodside
CA
94062
C1
Count Rudy Megawatt

1534 Mardi Gras Freeway
Fat City
LA
44061
C2
```

Figure 2a. *The data file to be converted; each record has seven lines, with each line ending in a Return (carriage return/line feed).*

You can select compiler options with the `O` selection from the main menu. In CP/M-80 systems the options display looks like this:

```
compile -> Memory
           Com-file
           cHn-file

Find run-time error  Quit
```

Usually the compile is to memory, the default setting. You can select `C` to create a ".COM" file holding the program (a ".COM" file can be executed directly from CP/M). Or you can create a ".CHN" file with the `H` option. A ".CHN" file can be run from another Pascal program using the supplied `CHAIN` procedure.

When you select either the `C` or `H` options, you get to decide where in memory you want the program to reside. You can skip this if your program is to run on the system you are using. These options are used more often to save programs that are to be run on another system with different memory requirements:

```
Start address: 1D76 (min 1D76)
End   address: DD53 (max E006)
```

```
Tony Bove,User's Guide,2055 Woodside Rd. #180,Redwood City,CA,94061,C1
Cheryl Rhodes,TUG Inc.,P.O. Box 5245,Redwood City,CA,94063,C2
Don Larchmont,TUG Concert Promotion,P.O. Box 0001,Woodside,CA,94062,C1
Count Rudy Megawatt,,1534 Mardi Gras Freeway,Fat City,LA,44061,C2
```

Figure 2b. *The file created by the program in figure 1, holding the converted data.*

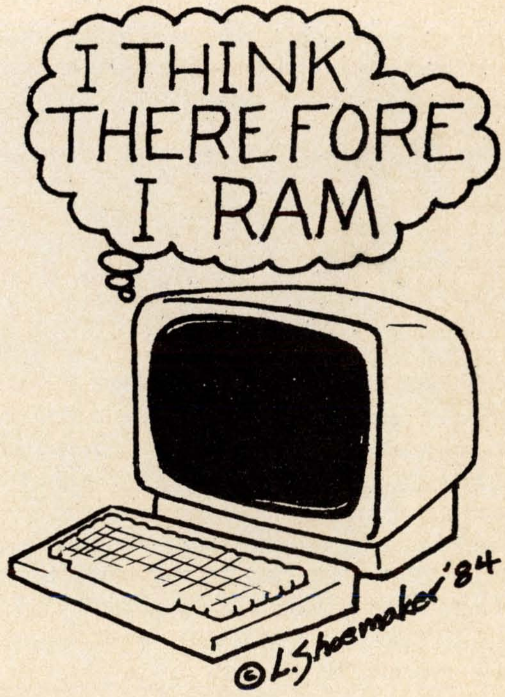
You can select another starting address as long as it is not lower than the minimum value. The addresses are in hexadecimal. The Pascal "run-time library" that supports your programs lies below the minimum address. To select another starting address, type `S` and the address in hexadecimal.

You can select another ending address if you wish. The address given is 700 to 1000 bytes below the BDOS module of CP/M-80, and the maximum value is one byte below the BDOS. The addresses are in hexadecimal. To select another starting address, type `E` and the address in hexadecimal.

In CP/M-86 you see a similar display with the `Cmd` option for ".CMD" files; in PC-DOS/MS-DOS you see a display just like CP/M-80. However in both cases you are allowed to change the assumed addresses for the CODE and DATA segments and the dynamic memory for your program.

When you run a Turbo Pascal ".COM" or ".CMD" program, and a run-time error occurs, the program displays a hexadecimal number next to `PC=`. Jot down this number, then use Turbo Pascal, define the work filename, select the `O` option for compiler options, and select the `F` compiler option to find the run-time error. Turbo Pascal asks for the `PC` (program counter) number. Type the number you jotted down, and Turbo Pascal will put the cursor on the line where the error occurred.

The next installment describes mathematical functions and structured programming.


I THINK THERE FORE I RAM
©L.Shoemaker '84

CP/M-80 •
CP/M-86 •
PC-DOS •
MS-DOS •

The CBASIC Compiler: CB-80 and CB-86

DRI's compilers for CBASIC are for serious programmers.

By Dale Blanchard

If you already program in BASIC and think you might want to write programs for the business market, you should take a good look at CBASIC, especially in its compiler forms, CB-80 and CB-86.

Most of us are familiar with the other main line of BASIC dialects, Microsoft BASIC, which comes in a wide variety of flavors beginning with the BASIC found on the Apple Computers and ending with the much-improved version found on the IBM PC. Microsoft BASIC is generally an interpreted BASIC, meaning that its statements are interpreted and executed immediately as you type them (one at a time).

Typesetting Conventions

We use `THIS typeface` to show exactly what you should type at your keyboard, with the special symbols ^ for the Control key (CTRL, CNTL or ALT on some keyboards), and ↵ for the Return key (RET, CR or Carriage Return on some keyboards). We use `This Typeface` to show messages displayed by the system or by programs.

CBASIC, originally designed by Gordon Eubanks as a business-oriented programming language, is now one of the family of programming languages from Digital Research, Inc. (DRI). In its original form, CBASIC is a semi-compiled BASIC, meaning that after you write your program, you process it through another program which interprets the program statements and puts them in what is called an intermediate file. This intermediate file is then "run" with still another program.

The primary advantage of compiling a program is speed. Once you have written and compiled your program, the compiled CBASIC version will almost always run faster than a Microsoft BASIC (MBASIC) version.

In 1981 DRI pushed the speed of CBASIC up another notch by taking the compilation one step further. Programs written in CBASIC can now be compiled into a form which can be run directly by the computer without having an intervening "run program." The version of CBASIC which produces fully compiled programs is known as CB80 and CB86, or the "CBASIC Compiler".

The CBASIC Compiler is designed for the serious programmer. DRI has done a good job of protecting your time

The primary advantage of compiling a program is speed. Once you have written and compiled your program, the compiled CBASIC version will almost always run faster than a Microsoft BASIC version.

CP/M and CBASIC are registered trademarks, and CB-80 and CB-86 are trademarks of Digital Research, Inc.

MBASIC is trademark of Microsoft.

investment by making the CBASIC Compiler available under CP/M-80, CP/M-86, CP/M-68K, MS-DOS, and PC-DOS. What does this mean for you (if you're a serious programmer)? It means that to make your programs run on the other operating systems, all you have to do is re-compile the "source code" of your program under the new operating system and your program is ready to go.

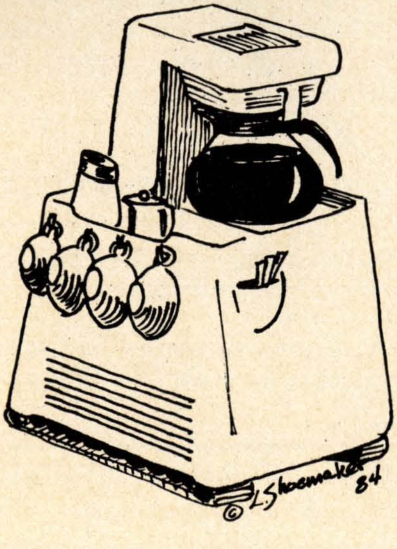
In the PC-DOS and MS-DOS operating systems, the compiler creates ".EXE" files. In eight-bit CP/M systems, the compiler creates ".COM" files, while in the sixteen-bit version it creates ".CMD" files.

With all that as background, let's look at some of the things I particularly like about the CBASIC Compiler and how I use the CBASIC language. Before we go any further, however, you should know that I assume you already know something about programming in BASIC of one dialect or another and don't need standard concepts like GOTO, GOSUB, RETURN, WHILE, WEND, or INPUT explained.

Multi-line Functions and Line Numbers

In CBASIC you can define your own multi-line functions. Multi-line functions are much like standard BASIC subroutines with some advantages which I will get to later. They must begin with a definition statement (DEF) and must end with a function-end statement (FEND). Between the definition and the function end, you can do almost anything you want including GOTO AND GOSUB as long as you stay within the function. That is; you can't GOTO or GOSUB outside the function.

The Latest in Office Automation



The CoffeeBot

Functions can be as complex or as simple as you want, from several lines to just one. Here is a relatively simple one I include in almost all of my programs:

```
DEF FN.CONTINUE% PUBLIC
        REM ****************************************
        REM ***** PRESS ANY KEY TO CONTINUE *****
        REM ****************************************
        STOP.PRINT% = 0
        PRINT ''PRESS ANY KEY TO CONTINUE '';
        WHILE NOT CONSTAT%
        WEND
        I% = INKEY
        IF I% = 27 THEN STOP.PRINT% = 1
FEND
```

I use FN.CONTINUE% to end printouts of error messages and to interrupt other print-outs on the computer screen. Notice that I used no line numbers. In CBASIC programs it is not necessary to use line numbers unless they are referenced with a GOTO or GOSUB statement. Here is a short program fragment which further illustrates this as well as illustrating the use of functions.

```
100 PRINT ''CHOOSE A NUMBER BETWEEN 1 & 10 '';
        NN$ = FN.INPUT$
        IF FN.NUMBER%(NN$) = 0 THEN\
                        CALL FN.BELL%:\
                        CALL FN.ERR1%:\
                        CALL FN.CONTINUE%:\
                        CALL FN.CLR%:\
                        GOTO 100
```

There are several functions here that do various things:

(1) FN.INPUT$ is my own input function so I don't have to rely on the built-in BASIC input functions.

(2) FN.NUMBER%(NN$) is a function which checks the input to see if it is a number.

(3) FN.BELL% is a function which clears the screen, rings the bell, and positions the cursor half-way down the screen where I want my error message printed.

(4) FN.ERR1% is an error message telling the operator that the input is not a valid number.

(5) FN.CONTINUE% we already know about.

(6) FN.CLR% clears the screen before going back to line 100 to re-prompt the operator.

Of course, all of this could have been accomplished with subroutines using the GOSUB statement, but then we would have to remember the line number of each subroutine every time we wanted to use it. I think it's easier to remember names than numbers. Another advantage of functions is that they can be defined as a public functions so they can be called from overlays. You can't do that with subroutines.

And now a final thought about line numbers: they do not have to be in order; in fact, they don't even have to be numbers. `GOSUB GET.TITLE` is a valid CB80 statement.

The LPRINTER Statement

This is the next feature of CBASIC which sets it apart from the various forms of Microsoft BASIC. In Microsoft BASIC if you want to give the operator a choice of screen or printer printouts, you have to do something like this:

```
10 PRINT ‘‘Where do you want your printout?’’
20 INPUT ‘‘On the printer or on the screen? (P/S) ’’;NN$
30 IF NN$ = ‘‘S’’ THEN GOSUB 60
40 IF NN$ = ‘‘P’’ THEN GOSUB 100
50 GOTO 10
60 REM ***** SCREEN PRINTOUT *****
65 PRINT ‘‘Hello, there.’’
70 PRINT ‘‘How are you doing?’’
80 RETURN
100 REM ***** PRINTER PRINTOUT *****
105 LPRINT ‘‘Hello, there.’’
110 LPRINT ‘‘How are you doing?’’
120 RETURN
```

That is, you have to write two subroutines for each printout. That results in long, memory- and storage-eating programs.

In CBASIC you would write the program this way:

```
10 PRINT ‘‘Where do you want your printout?’’
   INPUT ‘‘On the printer or on the screen? (P/S) ’’;NN$
   IF NN$ = ‘‘P’’ OR NN$ = ‘‘S’’ THEN GOSUB 60
   GOTO 10
60 REM ****************************
   REM ***** PRINTOUT ROUTINE *****
   REM ****************************
   IF NN$ = ‘‘P’’ THEN LPRINTER
   PRINT ‘‘Hello, there.’’
   PRINT ‘‘How are you doing?’’
   CONSOLE
   RETURN
```

You write only one subroutine and use the LPRINTER and CONSOLE statements to switch back and forth between the printer and the screen. In programs with several long printout routines, this can be a tremendous advantage.

The UCASE$ Function

CBASIC has a built-in function called UCASE$. In the above example our program would work fine as long as the operator was typing in all UPPER CASE letters. If not, the program would not respond. The Microsoft BASIC program should be written like this:

```
10 PRINT ‘‘Where do you want your printout?’’
20 INPUT ‘‘On the printer or on the screen? (P/S) ’’;NN$
30 IF NN$ = ‘‘S’’ OR NN$ = ‘‘s’’ THEN GOSUB 60
40 IF NN$ = ‘‘P’’ OR NN$ = ‘‘p’’ THEN GOSUB 100
```

That is; we have to entertain the possibility of either UPPER CASE or lower case input.

In CBASIC we would handle the problem this way:

```
10 PRINT ‘‘Where do you want your printout?’’
   INPUT ‘‘On the printer or on the screen? (P/S)’’;NN$
   NN$ = UCASE$(NN$)
```

That tells the program that no matter what the operator types in, convert it to UPPER CASE before deciding what to do with it. That may seem like a small thing, but I have seen a lot of programs which refuse to respond to lower-case inputs.

Number Handling

CBASIC also does a good job with numbers. In Microsoft BASIC you normally get six digits of precision. Numbers larger than that are rounded off. For example, $10,000.99 would be rounded off to $10,001.00. For business purposes that is not close enough.

True, you can tell Microsoft BASIC to handle the $10,000.99 as a double-precision number, and you would get sixteen digits of accuracy. That, however, further slows down an already slow program. CBASIC automatically and quickly gives you fourteen digits of accuracy.

Summary

Here's a summary of the advantages of compiled CBASIC over Microsoft BASIC and other BASICs like Microsoft's:

(1) CBASIC programs run faster.

(2) You can define your own multi-line functions.

(3) You don't have to use line numbers. This results in more structured, easier to read programs. Another advantage is that you can insert new lines into your program without having to worry about a line number being out of order.

(4) The LPRINTER statement. This allows writing only one subroutine for printouts and then switching back and forth between the screen and the printer.

(5) The UCASE$ function. This allows simpler checking of operator inputs.

(6) The accuracy of its number handling. For serious business programs that is absolutely essential.

Function Examples

Below are some of the functions I use in my programs along with some explanation of what they do.

```
DEF FN.NUM%(SUM$) PUBLIC
     REM ****************************************
     REM ***** CHECKS INPUT STRING *****
     REM ***** FOR NON-NUMERICAL ENTRIES *****
     REM ***** SUM$ WILL ALWAYS BE A STRING *****
     REM ****************************************
     DEC% = MATCH(''.'',SUM$,1)
     IF DEC% = 0 THEN WHOLE$ = SUM$:GOTO NUM1
     WHOLE$ = LEFT$(SUM$,(DEC%-1))
     DECIMAL$ = RIGHT$(SUM$,LEN(SUM$)-LEN(WHOLE$)-1)
NUM1: IF LEN(WHOLE$) = 0 THEN GOTO NUM2
     IF      WHOLE$ = ''-'' THEN GOTO NUM2
     IF LEFT$(WHOLE$,1) <> ''-''\
       AND LEFT$(WHOLE$,1) <> ''0''\
       AND VAL(LEFT$(WHOLE$,1)) = 0 THEN GOTO NUM4
     IF LEN(WHOLE$) < 2 THEN GOTO NUM2:
     FOR C% = 2 TO LEN(WHOLE$)
       IF MID$(WHOLE$,C%,1) <> ''0''\
        AND VAL(MID$(WHOLE$,C%,1)) = 0 THEN GOTO NUM4
     NEXT C%
NUM2: IF LEN(DECIMAL$) = 0 THEN GOTO NUM3
     FOR C% = 1 TO LEN(DECIMAL$)
       IF MID$(DECIMAL$,C%,1) <> ''0''\
        AND VAL(MID$(DECIMAL$,C%,1)) = 0 THEN GOTO NUM4
     NEXT C%
NUM3: FN.NUM% = 1:\
     RETURN
NUM4: FN.NUM% = 0
     FEND
```

With this function we are asking the program to check the operator's input to see if it is a real number. To do that we ask the function to evaluate a string variable called SUM$.

The function first looks for a decimal point, and if it finds one, it breaks the string SUM$ into two parts, WHOLE$ and DECIMAL$. Then the function looks for a leading minus sign and checks each character of the two parts of the string to determine if it is a number. If the function finds a character which is not a number, the program jumps to NUM4 which assigns the function a value of zero, meaning we have an invalid number.

You would use this function like this:

```
10 INPUT ''Put in a number '';SUM$
   IF FN.NUM%(SUM$) = 0 THEN\
      PRINT BELL$:\
      GOTO 10
```

This function is useful at the end of error messages or before print operations:

```
DEF FN.CONTINUE% PUBLIC
   REM *************************************
   REM ***** PRESS ANY KEY TO CONTINUE *****
   REM *************************************
   STOP.PRINT% = 0
   PRINT ''PRESS ANY KEY TO CONTINUE '';
   WHILE NOT CONSTAT%
   WEND
   I% = INKEY
   IF I% = 27 THEN STOP.PRINT% = 1
   FEND
```

This function is used at the end of error messages and between pages on print-outs. The `WHILE NOT CONSTAT%` statement tells the program that if no key has been pressed, sit and wait for one. If the key that is pressed is the escape key, then set `STOP.PRINT%` to 1. You would use this function like this:

```
90 REM **************************
   REM **** PRINTOUT ROUTINE ****
   REM **************************
   FOR X% = 1 TO END.OF.TEXT%
   PRINT A$(X%)
   LINE.COUNT% = LINE.COUNT% + 1
   IF LINE.COUNT% >= PAGE.LENGTH% THEN CALL FN.CONTINUE%
   IF STOP.PRINT% = 1 THEN RETURN
   NEXT X%
   PRINT ''PRINTOUT FINISHED''
   CALL FN.CONTINUE%
   RETURN
```

The following function interrupts a print operation:

```
DEF FN.INTERRUPT% PUBLIC
REM ******************************
   REM ***** PRINTOUT INTERRUPT *****
   REM ******************************
   STOP.PRINT% = 0
   I% = INKEY
   IF I% = 27 THEN FN.INTERRUPT% = 1:\
     RETURN
   WHILE NOT CONSTAT%
   WEND
   I% = INKEY
   IF I% = 27 THEN FN.INTERRUPT% = 1 ELSE\
                FN.INTERRUPT% = 0
   FEND
```

This function is used to cause a print operation to pause in the middle by pressing any key. The function first checks to see what key was pressed to cause the function to be called. If that key was the escape key, then the operator must have wanted to abort the print operation. If not, the function waits for another key to be pressed. You might use this function like this:

```
90 REM **************************
   REM **** PRINTOUT ROUTINE ****
   REM **************************
   FOR X% = 1 TO END.OF.TEXT%
```

```
PRINT A$(X%)
IF CONSTAT% THEN STOP.PRINT% = FN.INTERRUPT%
IF STOP.PRINT% = 1 THEN RETURN
NEXT X%
PRINT ''PRINTOUT FINISHED''
CALL FN.CONTINUE%
RETURN
```

The following three functions are screen handling functions and illustrate that functions can call other functions (both FN.BELL% and FN.NO.BELL% call FN.CLR% and FN.DOWN%). The fourth function is a "printer ready" function:

```
DEF FN.DOWN%(I%) PUBLIC
   REM ***********************************
   REM ***** LOWER CURSOR I% LINES *****
   REM ***********************************
   PRINT STRING$(I%,DOWN$)
   FEND

DEF FN.BELL% PUBLIC
   REM *********************************************
   REM ***** CLEAR THE SCREEN, RING THE BELL *****
   REM ***** PUT THE CURSOR HALF-WAY DOWN *****
   REM *********************************************
   CALL FN.CLR%
   PRINT BELL$
   CALL FN.DOWN%(10)
   FEND

DEF FN.NO.BELL% PUBLIC
   REM *********************************************
   REM ************ CLEAR THE SCREEN *************
   REM ****** PUT THE CURSOR HALF-WAY DOWN *******
   REM *********************************************
   CALL FN.CLR%
   CALL FN.DOWN%(10)
   FEND

DEF FN.READY% PUBLIC
   REM **********************************************
   REM ***** PROMPT OPERATOR TO CHECK PRINTER *****
   REM **********************************************
   PRINT ''IF PRINTER IS READY...''
   FEND
```

Here's an example using the above four functions:

```
100 CALL FN.PRINT.DEST%       REM   GET   DESTINATION   OF
                              PRINTOUT
   IF PRINT.DEST$ = ''2''
THEN\                         REM IF PRINTOUT ON PRINTER...
     CALL FN.NO.BELL%:\       REM CLEARS THE SCREEN
     CALL FN.READY%:\         REM OPERATOR CHECKS PRINTER
     CALL FN.CONTINUE%        REM PRESS ANY KEY TO CONTINUE
   GOSUB PRINT.OUT.ROUTINE REM DO THE PRINTOUT
   RETURN                     REM RETURN TO MAIN MENU
```

In a short program there would be no particular advantage to defining all these routines as functions rather than as subroutines, but in a large program with several overlays, the advantages are enormous. By defining the functions as PUBLIC, they can be called from any part of the program. Also, since they are functions, you don't have to remember what line number they are on, only what their names are.

Another nice feature: these functions can be kept in a library file and linked to any program you write. Thus, they have to be written only once, and can then be used over and over in different programs.

If you are interested in a more detailed discussion of user-defined functions in CBASIC take a look at *CBASIC User Guide* by Adam Osborne and Gordon Eubanks (Eubanks is the original author of CBASIC).

Wish List

As you have probably gathered, I think the compiled version of CBASIC is a powerful programming language. I have watched it evolve from version 1.3 to its present 2.0 form. It has come a long way, but there are some things I would like:

1. I wish it had a file-print function, FPRINTER, just like LPRINTER. Then we could use the same subroutine to print on the screen, on the printer, or into a disk file.

2. I fervently wish it had a function similar to Microsoft's FILES command. I don't see why we should have to exit from a program to look at the disk directory.

3. I wish it had a function to tell me how much disk space still remains. It would be nice to be able to warn people that they were about to run out of disk space rather than simply let them know afterward.

4. Now that we've moved into the sixteen bit world with all that memory being available, we've got to have a way of measuring it. Right now we can still only measure 64k.

I have discussed these wishes with Digital Research and it looks as if at least some of them will be granted in the near future:

Wish #1 is apparently in the works.

Wish #2 is a different matter. I had a great deal of trouble getting the person I was corresponding with to understand why anyone would want to look at their disk directory from within a program. I don't know if I made any headway or not.

Wish #3 was at least understood and may be in the works.

Wish #4 is definitely in the works.

CP/M-80 •
CP/M-86 •

From Source To Object: Executable Programs

How to use ASM or ASM-86, with LOAD or GENCMD, to produce executable assembly language programs, even without training in assembly language.

Are you missing opportunities to try programs because they are in source code form, and you don't know what to do with them because you don't want to have to learn about assembly language? Try this brief tutorial in "assembling a program," written for either CP/M-80 *or* CP/M-86 (don't bother reading both unless you have both systems).

Why do you need this information? You might have what's called a "source file" version of a program that needs to be "assembled and loaded" in order to run on your system. This article explains how to do that.

Or you may want to make a "patch" to your system or to your programs. All versions of CP/M are supplied with utility programs for performing program assemblies and "patches." You can write entire programs in assembly language with these tools.

Utilities for CP/M-80 systems include ASM (an "assembler" of program statements), DDT (a "debugger" of programs, used also for "patches"), and LOAD (for producing a ".COM" program). Their counterparts in CP/M-86 are ASM-86, DDT-86 and GENCMD. A DDT tutorial appeared in issue #8 of *User's Guide.*

Assembling a Program

If you intend to write an assembly language program, get a book on assembly language. Assembly language is perhaps the most difficult programming language to use, but anyone can *edit* a program written in assembly language, and perform the operation of *assembling a program.*

Such programs start their lives as text files created by a text editing program like ED (supplied free with CP/M) or word processing program like WordStar® (MicroPro). Someone with minimal training can open a text file with ED or WordStar and change its contents. You may not understand the text in the file, because it is written in assembly language, not English. But given the appropriate instructions for what to change, you can perform the change as you would edit any text file.

Assembly language text files are called *source files.* Source files written for eight-bit systems (CP/M-80) use

Typesetting Conventions

Throughout this article we use `THIS typeface` to show exactly what you should type at your keyboard, with the special symbols ^ for the Control key (CTRL, CNTL or ALT on some keyboards), and ↵ for the Return key (RET, CR or Carriage Return on some keyboards).

Since computer activities nearly always involve typing, you'll see `THIS typeface` used a lot to show commands, with letters in all upper case to distinguish commands from regular text. You may not have to type in upper case — we use it to emphasize commands. We also use `lower case` for text in text files and for text that varies with the specific command (e.g., `DIR filename` tells you to type the `DIR` command with any filename).

We also use `This Typeface` to show messages displayed by the system or by programs, and to show older text in examples where you are to type new text or commands.

by Tony Bove & Cheryl Rhodes

filenames that have ".ASM" as the common extension. Source files written for sixteen-bit systems (CP/M-86) use filenames that have ".A86" as the common extension. You can name your files as you wish, but you must use these conventions to make them work in the following examples.

The procedures differ for CP/M-80 (eight-bit) and CP/M-86 (sixteen-bit) systems, so we separate the instructions.

CP/M-80: Assemble & Load

Assembly language source files can be turned into working programs by *assembling* them with the ASM utility and *loading* the programs with the LOAD utility. The two operations produce a program that can run on almost any computer using the 8080, 8085 or Z80 processors.

Assuming you've made the changes you need to make to the source file and you're ready to assemble it, and assuming your source file is named SAMPLE.ASM and located in drive A, the command you type to assemble it is:

```
A>ASM SAMPLE↵
```

The ASM utility produces two new files on the disk in drive A: SAMPLE.HEX and SAMPLE.PRN. The first is the important one for creating a program — it is called the *object file* for the program. It contains the program in hexadecimal code for use with the LOAD utility. The second is a printable listing of the program, which takes up more disk space than the first one.

If you run out of space on disk, or run out of directory entries, you will get one of these error messages: `OUTPUT FILE WRITE ERROR` (no space left) or `NO DIRECTORY SPACE` (no room for new filenames).

You can use other drives for these files by adding a special extension to the filename on the ASM command line to control where to find the ".ASM" source file and where to put the ".HEX" and ".PRN" files. The extension has three letters:

```
A>ASM SAMPLE.ABX↵
```

The first character (`A`) tells ASM to look in drive A for the source file (".ASM" file). The second (`B`) tells ASM to put in drive B the object file (".HEX" file). The third character (`X`) tells ASM to display the listing rather than create a listing file (".PRN" file). This saves space on the disk, and you can print the listing if you type Control-`P` before typing the ASM command line (Control-`P` turns on printer "echo" of the screen).

To summarize, here's the generic form of the ASM command line:

```
ASM filename.ahp↵
```

Substitute your own filename without the ".ASM" extension (ASM assumes all source files have ".ASM" as the extension), and use the extension to tell ASM which drive to look into for the ".ASM" file (`a`), which drive to store the ".HEX" file (`h`), and which drive to store the ".PRN" file (`p`). You can substitute a `Z` for `p` to cancel the ".PRN" file, or `X` to display it first on the screen and then cancel it.

After assembling the ".ASM" file to produce the ".HEX" file, your final step is to produce the ".COM" file using the LOAD utility (LOAD.COM). Assuming your ".HEX" file was produced from SAMPLE.ASM (SAMPLE.HEX), you would type this command (assuming also that SAMPLE.HEX and LOAD.COM are on drive A):

```
A>LOAD SAMPLE↵
```

This produces SAMPLE.COM, which can then be executed by typing `SAMPLE` and pressing Return (↵). The ".COM" file is a directly-executable CP/M program, which means you can run the program by typing its name as a command. The ".HEX" file is not directly executable, but it can be loaded into memory and run using the DDT utility, which is shown later.

CP/M-86: Assemble & Generate

Assembly language source files can be turned into working programs by *assembling* them with the ASM-86 utility, and producing executable programs with the GENCMD utility. The two operations produce a program that can run on almost any computer using the 8086, 8088, iAPX 186/286 or iAPX 188/288 processors.

Assuming you've made the changes you need to make to the source file and you're ready to assemble it, and assuming your source file is named SAMPLE.A86 and located in drive A, the command you type to assemble it is:

```
A>ASM86 SAMPLE↵
```

The ASM-86 utility produces three new files on the disk in drive A: SAMPLE.H86, SAMPLE.LST and SAMPLE.SYM.

SAMPLE.H86 is the important one for creating an executable program — it is called the *object file* for the program. It contains the program in hexadecimal code for use with the GENCMD utility.

SAMPLE.LST is a printable listing of the program, which takes up more disk space than the first one and may not be needed.

SAMPLE.SYM is a *symbol file* used when "debugging" a program (something you are not likely to do without first reading about assembly language).

If you run out of space on disk, or run out of directory entries, you will get one of these error messages: `Disk Full` (no space left) or `Directory Full` (no room for new filenames).

You can use other drives for these files by adding *parameters* to the ASM-86 command line following a dollar sign:

```
A>ASM86 SAMPLE $AA HB PX↵
```

The first parameter (`AA`) tells ASM-86 to look in drive A for the source file (".A86" file). The second parameter (`HB`) stores the the object (".H86") file in drive B. The third parameter (`PX`) displays the listing rather than create a listing (".LST") file. This saves space on the disk, and you can print the listing if you type Control-`P` before typing the ASM-86 command line (Control-`P` turns on printer "echo" of the screen).

To summarize, here's the generic form of the ASM-86 command line:

```
ASM86 filename $ Ad Hd Pd Sd FI ↵
```

Substitute your own filename with any extension, or an ".A86" file's name without an extension (ASM-86 assumes the ".A86" extension). Type optional parameters following a dollar (*$*) sign to control the files as follows:

`Ad`	Look for source (".A86") file on drive `d`.
`Hd`	Store the hexadecimal object (".H86") file on drive `d`.
`Pd`	Store the listing (".LST") file on drive `d`. Type `X` for `d` to display the listing rather than store it in a disk file; or type `Y` to print the listing rather than store it; or type `Z` to skip it altogether.
`Sd`	Store the symbol (".SYM") file on drive `d`. Type `X` for `d` to display the symbol file rather than store it in a disk file; or type `Y` to print the symbol file rather than store it; or type `Z` to skip it altogether.
`FI`	Produce a hexadecimal object file using the Intel format; otherwise, ASM-86 uses the default Digital Research format.

After assembling the ".A86" file to produce the ".H86" file, your final step is to produce the ".CMD" file using the GENCMD utility (GENCMD.CMD). Assuming your ".H86" file was produced from SAMPLE.A86 (SAMPLE.H86), you would type this command (assuming also that SAMPLE.H86 and GENCMD.CMD are on drive A):

```
A>GENCMD SAMPLE ↵
```

This produces SAMPLE.CMD, which can then be executed by typing `SAMPLE` and pressing Return (↵). The ".CMD" file is a directly-executable CP/M-86 program, which means you can run the program by typing its name as a command. The ".H86" file is not directly executable, but it can be loaded into memory and run using the DDT-86 utility, which is shown later.

GENCMD has optional parameters you can use in the command line to produce programs that are similar in nature to CP/M-80 programs, or that have certain memory requirements. The optional parameters follow the filename in the command line. Here's a quick summary of the parameters:

```
8080
```

The keyword 8080 tells GENCMD to generate a program (".CMD" file) that closely resembles an eight-bit CP/M-80 program that fits within a 64K memory space. This type of program has overlapping code and data areas and starts at memory address 0100H within the 64K memory space. This parameter is often used to generate a program that was converted from a CP/M-80 version to run with CP/M-86.

```
CODE[An,Bn,Mn,Xn]
DATA[An,Bn,Mn,Xn]
STACK[An,Bn,Mn,Xn]
EXTRA[An,Bn,Mn,Xn]
```

The keywords define *segment groups* (areas of memory associated by a relationship) that have specific memory requirements, which you can spell out within the brackets using the parameters with `n`, which in all cases is a paragraph boundary (paragraphs are 16 bytes long).

Within the brackets associated with any or all of the above keywords, use the following where needed:

`An`	Load a group at an absolute location in memory (paragraph boundary `n`).
`Bn`	Begin a group at address `n` in the hexadecimal file. Usually this parameter is not needed if you used ASM-86 to produce the hexadecimal (".H86") file and chose the default Digital Research format. If you used the Intel format, you will need this.
`Mn`	The group requires a minimum of `n` paragraphs (`n` times 16 bytes). Use this when you include a data segment in the program that has an uninitialized data area at the end of the segment.
`Xn`	The group can use up to `n` paragraphs (`n` times 16 bytes) more than the minimum defined by `Mn`. Use this if your program can use a larger data area.

Here are two examples of GENCMD command lines — the first generates a CP/M-86 program that is similar in organization to a CP/M-80 program, and the second generates a CP/M-86 program with specific instructions for the CODE and DATA groups:

```
A>GENCMD FIRSTPRG 8080 ↵
A>GENCMD SECONDPR CODE[A50] DATA[M40,XFFF] ↵
```

Variations of GENCMD command lines are possible, but not likely to be used unless you are doing special programming. If you are given a source file to generate into a program, you would normally use the least complicated versions of ASM-86 and GENCMD, unless you are given specific instructions on using the parameters.

MS-DOS and PC-DOS, Part III

An introduction to MS-DOS commands from a CP/M user's point of view.

The commands that an operating system provides for you are your set of everyday user's tools — if they're a good set, they can make your interaction with the computer much easier and more productive.

If you're a CP/M user, by now you're probably familiar with at least some of the more common CP/M commands and utilities. At a minimum, you probably use DIR to see directories, and PIP to copy files. Microsoft's MS-DOS operating system has its own set of commands and utilities, but they shouldn't be that hard to learn. Like the rest of MS-DOS (and IBM's PC-DOS version), this 16-bit operating system's commands and utilities clearly show their CP/M heritage.

As with the rest of the two systems, many of the differences in the command sets is due to the difference in intended audience and the changing economies of hardware. CP/M was originally written for program developers, in an era when hardware was more limited and more expensive. MS-DOS, in contrast, was intended for the computer user, one who can afford to buy more memory rather than spend more time.

PC-DOS/MS-DOS®

MS-DOS (Microsoft) is the operating system that is widely associated with the IBM PC and PC-compatible computers. PC-DOS, which runs on the IBM PC, is a version of MS-DOS that has some unique features not in MS-DOS. Unless otherwise specified, everything said about MS-DOS applies to PC-DOS (the version of MS-DOS used on the IBM PC series). Anything said about PC-DOS does not necessarily apply to the more generic MS-DOS.

by Steve Rosenthal

Preparing your Disks

The first task in using either a CP/M or MS-DOS computer should be to make back-up disks. Both systems offer a set of utilities for creating and copying disks.

Before you can use disks on either system, you have to initialize or format them — this process writes down the precise pattern of magnetic tracks that serve as the framework for storing and locating information. On most CP/M systems, initializing disks is done with a program named FORMAT. You use SYSGEN, a separate program, to put a copy of CP/M itself on formatted disks. Disks generally have to be SYSGENed to be used in drive A.

MS-DOS has a similar FORMAT command, but as an option it will also write the system onto the new disk. This allows you to format the disk and transfer the system in one step. MS-DOS also has a SYS command to transfer DOS to a new disk, but it's used almost exclusively to put your own copy of DOS on copy-protected store-bought software.

The second step in backing up your original disks is to copy the files. In CP/M, the copy command is PIP (short for Peripheral Interchange Program). PIP will copy any single file or group of files from one disk to another.

You must have PIP on one of the current disks in order to use it — like most of the more complex commands in CP/M, using the command means running a program from a disk file (see the accompanying box on Internal Commands).

To most users, PIP does seem to have some peculiarities. Because of the disk-change checking procedures within CP/M, you can't put in a different disk in a destination drive during the PIP program, but you can change the source disk. When you specify a transfer, the syntax is PIP to the first file from a second, rather than the more intuitive forward direction from the first file to the second.

In MS-DOS, you can transfer your files to a new disk with the COPY command. COPY is an internal command, so it is always accessible and quick to use. You can switch disks and continue copying whenever you get a prompt. Unlike PIP, COPY only transfers files — it does not have PIP's extensive filtering and editing capability.

If you're copying entire disks and not selected programs, MS-DOS offers a faster method with the DISKCOPY command. When you use this command, it checks to see if the destination disk is properly formatted, and if not formats it as it copies. Copying is done track by track, including the system files if they're on the source disk.

Some CP/M systems also have a similar one-step copy program. On some machines the disks must be formatted first, while on others the disk copy command will arrange for any necessary formatting as well.

If you have any doubts about the result of your MS-DOS disk copying, you can use the DISKCOMP command to see if the resulting disk is identical to the original. Note that if you move the files over with the COPY command, chances are that the arrangement of the data may be different on the new disk so it may show as different in the DISKCOMP program. If you use DISKCOPY, the disks should test as perfect matches.

If you have been transferring files with the COPY command, you can still check them, but you use the COMP command instead of DISKCOMP. This command checks to see if files on different disks are identical on a file-by-file basis.

Seeing What You Have

Theoretically, making backup copies of system disks or new programs should come before anything else. In practice, most of us take a peek at what we have before starting the backup process.

The most direct way to see what is on a disk in CP/M is to use the DIR (show the directory) command. This command gives you a list of files, with options to show files on another drive or matching a specified pattern.

If you want a more detailed display, you have to use the STAT command (or in CP/M 3.0, the extended DIR command). These more complete displays require external programs, so they're slower and require you to use some of your disk space for the program.

MS-DOS has a corresponding DIR command, but it provides a slightly different display. Because MS-DOS has file length information in each on-disk directory entry (see Part II of this series in UG Vol 2.2 No. 8 for details), the MS-DOS directory display shows the length in bytes for each file.

If you want further information about the CP/M directory, you use the STAT command (the extended DIR and SHOW commands in CP/M Plus). STAT shows you all or selected files on a disk in alphabetical order and with the correct size, plus statistics on disk usage and free space..

MS-DOS has no corresponding program. You do get file size with DIR, and versions starting with 2.0 automatically display free disk space. You can get some further disk usage statistics from the CHKDSK program, but not a sorted file display. However, you can get a sorted display using the SORT command and a pipe — a subject that we'll put off until the next installment.

Hierarchical Directories

In CP/M and MS-DOS versions before 2.0 for each system, directories were simple unstructured lists, so the plain DIR commands did just fine at showing you what you had. With the second generation of each system, the directory took on more complexity.

In CP/M, the added element was user areas. Each file entry was tagged with a value from 0 to 15 — to access a file, you had to have the system set to the matching user number.

MS-DOS took another route. Instead of going to the partitioned model used by CP/M, MS-DOS added *hierarchical* directories in version 2.0. In this method, directories can have entries that are *subdirectories*: files whose contents are themselves directories. Subdirectory entries can themselves have further subdirectories, and in theory so on until disk space is exhausted. In practice, more than a few levels quickly becomes unwieldy.

The particular hierarchical form taken by the MS-DOS directory is usually termed a "tree structure," because of the similar branching pattern. The main directory is thus termed the "root," and each subdirectory is a "branch." The final, bottom level (computer trees grow upside down) contains the leaves, the individual files. Note that the MS-DOS directory tree is not balanced or binary — each directory or subdirectory can have zero or many descendents, while leaves can sprout from the root as well as from branches.

In The Beginning...

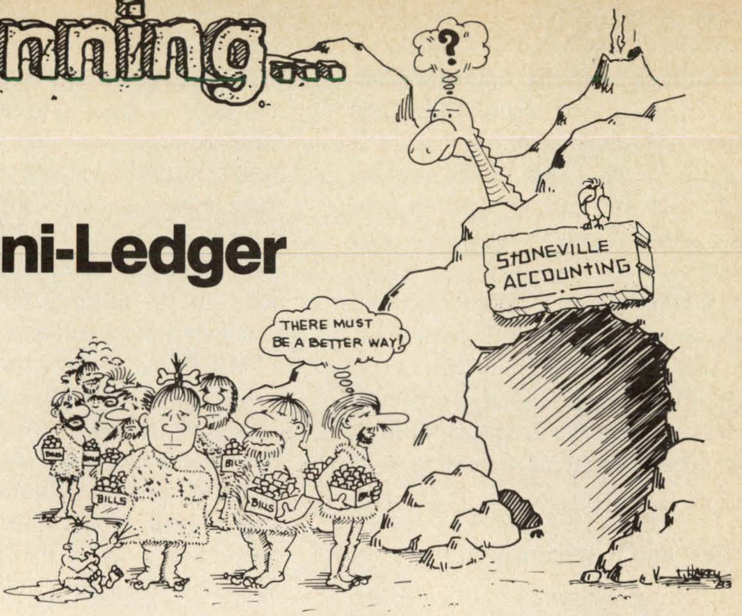
There was the "Shoe-Box" accounting system...

But Now There's Mini-Ledger

While those dinosaur-size companies resort to large, integrated computer accounting packages, your modest-size company can track key income and expense activities through the use of Paradigm's **single-entry Mini-Ledger** program. And you can do this for only $150.00.

With **Mini-Ledger,** you can define up to 99 codes like your company car expenses, entertainment, rent, payroll, cash flow...it's up to you. And because **Mini-Ledger** is so easy to use, you don't need any accounting experience to run it. Also, several printouts are available to help you prepare clear and accurate statements of your financial operation. That's ideal for tax purposes.

To employ **Mini-Ledger,** you need either CP/M™, CP/M-86, MS-DOS or PC-DOS 2.0, and at least one disk drive. A demo version with user's-manual is available for $30.00. *Substantial dealer discounts available.*



With Mini-Ledger, those prehistoric days of Shoe-Box accounting are over.

PARADIGM CONSULTANTS

MINI-LEDGER

39243 Liberty Street, Suite L
Fremont, CA 94538 **(415) 796-0543**
Outside California Call **(800) 821-8407**

CP/M is a registered trademark of Digital Research Corporation

Circle number 12 on the reader service card.

CP/M only needs one command to navigate through a partitioned directory. With the USER command, you specify which of the 16 user areas you want to operate in. If you don't specify an area, this command also displays your current user number value.

MS-DOS, by contrast, needs several special directory commands. To start with, you use MKDIR (which also works as MD) to create a new subdirectory. If you want to consider that new directory the home directory for future commands, you have to select it with the CHDIR (which also works as CD) command.

If you get lost, you can see your current directory by issuing a CHDIR command with no directory name. The system then prints out the name of the current directory, including any parent directories that stand between it and the root (main) directory.

Once you're finished with a subdirectory, you can't just erase it like you would any other file. The RMDIR (or RD) command removes directories, providing the directory is empty and you are not trying to remove the root (main) or current home directory.

MS-DOS also has a special hierarchical directory viewing command. In CP/M, when you issue a DIR command, you only see the files in the current directory (in CP/M Plus, you can see other user files with the extended directory command). In MS-DOS, you only see the files in the current directory.

To see the full MS-DOS directory, you use the TREE command. This program lists all subdirectories below the root, and optionally all file entries contained in those subdirectories.

Working with Files

CP/M comes with built-in commands to create, rename and erase files. You might never use the SAVE command to create a file (SAVE is intended for saving patched files after a session with the DDT or SID debugger programs), but you probably will use the others.

In CP/M, the ERA (erase) command is straightforward. Using the same syntax as the DIR command, any file that matches the specified name is removed from active status in the disk directory.

The REN (rename) command follows the common CP/M reverse syntax. You type `REN newname=oldname` ↵, rather than the more logical old to new.

MS-DOS, by contrast, has a small quirk in its erase command, but a clean implementation of the RENAME function. If you specify the ERASE command by its full name, it works almost identically to the CP/M equivalent. But if you abbreviate it as `ERA`, the system doesn't recognize it as the same command — instead, you have to remember to call it `DEL`.

RENAME, in MS-DOS, works as you would expect. The syntax is from old to new, so you type `REN oldname newname`. A nice addition is that MS-DOS allows wildcards in the rename function, so you can rename all programs with similar names in one command.

MS-DOS has no SAVE command. If you want to fool directly with memory, you can do it from within the DEBUG program. MS-DOS presumes that users have no frequent need to make files from memory images.

Setting the Environment

In CP/M, the assumption was that either you were a true hacker and adapted and installed the system code yourself, or you bought a complete system from a manufacturer who did the installation for you. That hasn't always been true, of course, and in CP/M Plus there has been some effort to offer ways to accommodate different hardware short of reassembling the whole code. But by and large, CP/M assumes that your system is constant.

MS-DOS, on the other hand, was written to let relatively unskilled users customize many important features. IBM wanted an open, expandable system that would accommodate growth, a wide family of machines, and support the add-on industry that made the PC adaptable to a greatly enlarged market segment.

Consequently, MS-DOS and particularly PC-DOS offer a broad range of environment-setting commands. At a cost of some extra overhead, the system gains a great deal of flexibility.

For example, in case your disk set up doesn't match what some program author expected, you can use the ASSIGN utility to redirect calls from one disk drive to another. This lets you use programs designed for floppies on a hard disk, or substitute drives without messing with protected disks or binary code.

If you want to be able to stop a program in progress even if it doesn't check the keyboard very often, you can set BREAK ON, forcing a keyboard check whenever a program asks DOS for any service. BREAK OFF restricts

---

Some Commands are Internal

Conceptually, MS-DOS, like CP/M before it, treats every command you give it as the name of a program. It takes the first word of your command line as an instruction to execute the program that has the command as its name.

For example, if you respond to MS-DOS's `A:` prompt with `DOBOOKS` and press Return, the system will search its disk files for a directory entry for file with a filename part called DOBOOKS and a file type of ".COM", ".EXE" or ".BAT". If MS-DOS finds such a program, it loads the program into memory and executes it. If not, the system tells you that it can't find such a file.

Some programs are so commonly used that operating system designers assume that you'll always want them available. So rather than put them on the disk as separate files, these short essential utilities are embedded within the operating system itself. Anytime you have a copy of the operating system, you have these utilities.

In CP/M, these commands are called the "built-in" commands. In MS-DOS, they're called "internal" commands. Both work the same way — before the system responds to your command by looking on the disk, it first checks to see whether your command can be found on its short list of included programs. If so, the system need not read the disk, because the requested program is already sitting in memory.

In CP/M, the built-in programs are:

`DIR`
Show the directory.
`ERA`
Erase files.
`REN`
Rename files.
`SAVE`
Save memory to a disk file.
`TYPE`
Type out a file on the console.
`USER`
Set or show current user area (2.2 and later).

With a larger address space, cheaper memory (and the expectation that users would have larger systems), MS-DOS included many more functions inside the DOS command interpreter. As of version 2.1, the MS-DOS internal commands are:

`BREAK`
Set break checking on and off.
`CHDIR`
Change your current directory.
`CLS`
Clear the screen.
`COPY`
Copy to and from files and devices.
`CTTY`
Change console device.
`DATE`
Request date.
`DIR`
Show disk directory.
`ERASE`
Delete files (used as DEL).
`MKDIR`
Create a subdirectory.
`PATH`
Set where DOS looks for files.
`RENAME`
Rename files (used as REN).
`RMDIR`
Remove a subdirectory.
`TIME`
Request the time.
`TYPE`
Type a file on the console.
`VER`
Show version number.
`VERIFY`
Set read-after-write on and off.
`VOL`
Show disk volume name.

checking to take place only during a more limited set of DOS calls.

If you feel your disk drives or disks are not goof-proof, you can set VERIFY ON to have the system read and compare each disk sector after it is written. On the other hand, if you want to speed operations and you trust your hardware, you can set VERIFY OFF.

In PC-DOS, the CTTY command (literally Change Teletype, meaning select a new console device) lets you switch the device the system will treat as the console from the usual keyboard and video adapter to another device. Although this is in ways more limited than CP/M's IOBYTE function in that it only redirects console I/O, it is not limited to the pre-defined devices: if you want, you can set up the system to treat a modem hooked up to the serial port as the console.

You can also customize PC-DOS with the GRAPHICS command if you want to be able to dump screen graphics to a standard IBM printer or equivalent. If you don't need this function, you don't have to invoke it — and you save 736 bytes in memory.

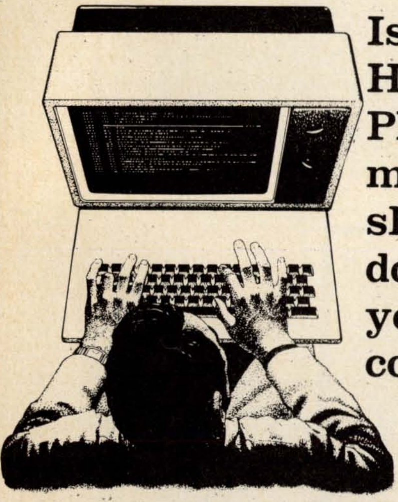
The ability to change the prompt line seems at first to be irrelevant or of little importance. It may seem to be only a ploy to make you feel like you are not totally under the control of the system. But if you use your computer for data logging or networking, or when you have users who are not expected to be familiar with the operating system, having a clear opening message can be a lot more useful than a cryptic `A:`. Using the PROMPT command, you can set up a multiline prompt, complete with time and date if you so choose.

You can also pass on parameters for use in individual programs with the SET command. This command acts like a message posting service, storing away data that application programs can check if they wish to. SET's main use is to let batch commands pass information on to programs, but you can use it directly if you know what information your programs might be looking for.

If your computer doesn't have a clock board that sets the time and date automatically, the TIME and DATE commands will ask the user to enter the correct values. Because MS-DOS machines are supposed to have at least interrupt-driven timers, once the time and date are set they should stay accurate until the computer is reset or turned off.

In PC-DOS, the MODE command takes care of many of the leftovers. This is the command you use to initialize the printer, set the video format, and set the parameters for the serial channels.

That's the first installment of MS-DOS commands from a CP/M point of view — check the next issue for an explanation of the batch commands and the way MS-DOS deals with hard disks.



Is the HUNT & PECK method slowing down your computing?

Master typing with . . .
TYPE-RITER the CP/M typing instruction and drill program

Menu Driven • Random Access Generation • 64,000 Lessons Scored Escape • Accurate Timing & Scoring • Records all Timed Test Scores • For Individual or Classroom Use

ORDER NOW — Call or Write
4418 Chapman, Suite 311
Orange, CA 92669
(714) 639-8106

LIST PRICE $29.95

CP/M is a registered trademark of Digital Research, Inc.

Circle number 13 on the reader service card.

FILEBASE™
The EASY-TO-USE data base manager for CP/M, PC-DOS, MS-DOS

Virtually all functions of programs costing hundreds more — for **just $100!** Demo disk/manual $10 (apply to purchase).

Generates and/or manipulates sequential or random access files (both with variable field lengths!) compatible with BASIC, WordStar, DataStar, dBase II, MailMerge and many others.

NO COMMANDS TO REMEMBER ... 100% menu & prompt driven. *No installation required!* Specify up to 40 fields with your own prompts in minutes. Change from sequential to random access in seconds.

"For its price, the best data base program we've seen"
User's Guide, Feb/Mar 84

"Everything you could ask for in documentation, function and price"
Dr. Dobb's Journal, Feb 84

"Sorts on last name or zip codes even if embedded in full name or city/state/zip field"
Microsystems, Aug 83

"Menu and prompt driven ... interactive record examination in real time"
Lifelines, Jun 83

"Above all, easy to learn, simple to use ... not a stripped down program."
R. Colwell in **Data Based Advisor**, Aug 84

"A powerful $100 product"
Info World 6/18/84

ALL FORMATS AVAILABLE: 8" CP/M and machine-specific 5¼" disks for CP/M, MS-DOS, PC-DOS (specify computer & operating system).

EWDP Software, Inc. P.O. Box 40283 Indianapolis IN 46240
AMEX, VISA, MasterCard, Check, COD. (317) 872-8799

Trademarks: CP/M, Digital Research; FILEBASE, EWDP Software; WordStar/MailMerge/DataStar, MicroPro; dBase II, Ashton-Tate.

Circle number 8 on the reader service card.

CP/M-80 •
CP/M-86 •
PC-DOS •
MS-DOS •

Batch Processing In CP/M and MS-DOS

Creating your own mini-programs to handle multiple operations.

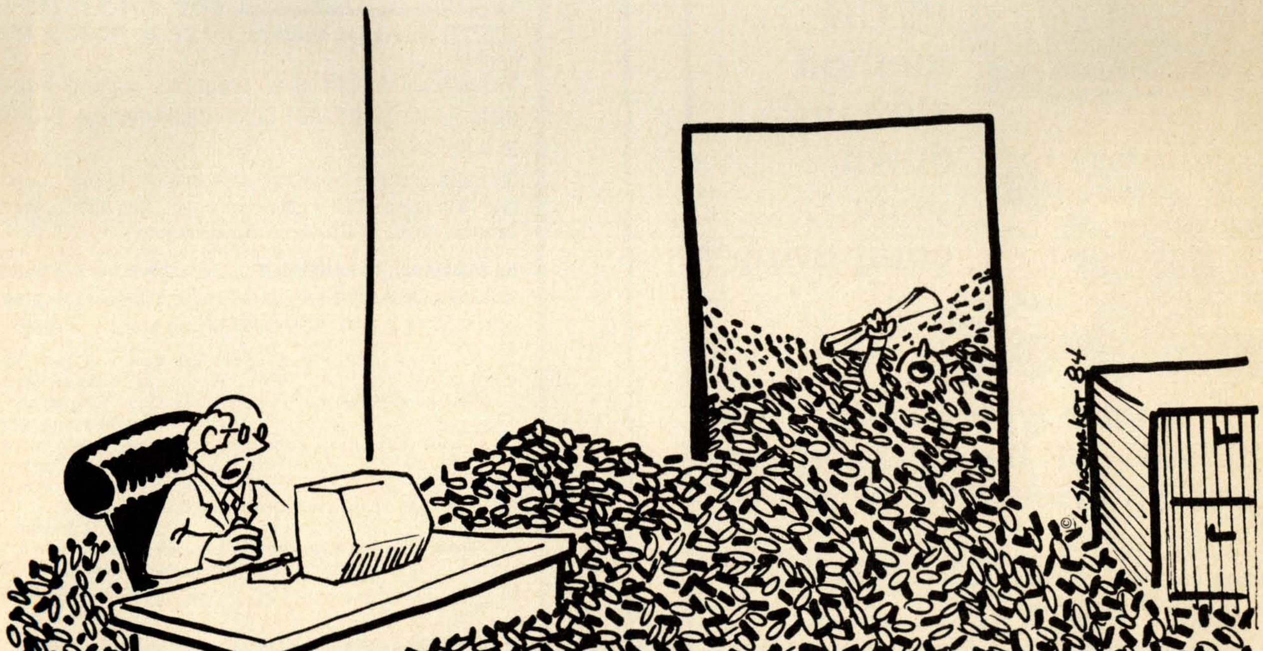
by Bob Wolff

Bob is the Product Manager for general ledger software at Tumbleweed Business Systems in Dallas, TX. He has a graduate degree in Accounting and is a self-taught computer user.

You've been working on a report for your boss or teacher for several weeks. You entered it into the word processor and spent days editing and rewriting it until it shines like a new wax job on a car. The night before it's due, you try to print it and get the disconcerting message `FILE NOT FOUND`.

A few muscles begin to tighten in your stomach. "Maybe I misspelled the document's filename," you tell yourself. You start searching through the directory. It is not there. Panic builds. You begin searching other user numbers of the hard disk and dozens of

The System Crash of 1984



Simple programming commands make MS-DOS batch files more powerful and flexible than CP/M batch files.

floppies. With each passing filename your dinner churns more. The chef salad feels more like jalepenos peppers. Sweat builds on your brow. As you approach the last disk directory, silent prayers are offered to a God you wish you talked with more often.

To anyone who has lost a file, whether created by a word processor, accounting program or a spreadsheet, this scenario is *very* real. You lived through a hell you swear will never happen again. If you've never lost an important file, you're either very lucky or have very well organized computer procedures.

Anytime important information is kept on a hard or floppy disk, a copy (called a "backup") should be made. Creating backup copies is relatively easy, but to keep them accurate and timely requires good work habits. These habits are not often learned until at least one stressful encounter with Murphy's Law.

One of the causes of poor backup habits is the boring, repetitive typing you need to do to copy files. It's not exciting to watch the filenames scroll by as they transfer to another disk. PIP and COPY programs produce multitudes of yawns.

A partial solution is available with the CP/M or MS-DOS operating systems (DOS). You can customize your copying operations for any type of files, whether produced by word processing or accounting programs. The solution is to use what are called *batch files*.

Batch Files

A *batch file* consists of text, just like a letter or a memo. You can create one with a fancy word processing program or a simple text editing program. The operating system (CP/M or MS-DOS) sees the text as a series of commands or programs. The commands are executed one after another in the sequence originally entered. You may also include comments to remind yourself or to explain to someone else what operations the batch file performs.

Batch files are easy ways to do repetitive jobs. Perhaps the most important job is file copying. The files may be documents or financial records or analyses; whatever the files, they must be safeguarded against the gremlins which inhabit imperfect technology.

You should also think about how to organize yourself so you can easily identify files and their contents without spending hours running them through the master programs. This is critical to effective safeguarding of important information.

If you don't think this is exciting, you're right. But you may need a copy of your business accounting records to show the IRS. Backing up your files properly can save thousands of dollars in time and lost income.

Batch files are easy to create. You only need limited experience with a word processing or text editing program to create a batch file. Each line of text in a batch file is either a command or program to be executed, or a comment to be displayed on the screen. Comments help you or someone else understand what you must do. It can also explain what is happening as each command is carried out.

Figures 1a and 1b show the same operation performed by batch files under CP/M (1a) and MS-DOS (1b). As you can see, the two batch files are very similar. Both have comments to explain what's happening. CP/M submit files use a colon (:) or a semicolon (;) to indicate a line is a comment. MS-DOS batch files use the word `REM` (remark) to indicate a comment. A comment displays on the screen (with possible variations under MS-DOS), then the next line executes.

A command like DIR (CP/M) or CHKDSK (MS-DOS) executes as if typed from the keyboard. Programs work the same way. After the command or program executes, the next line is carried out. This process continues until each line in the batch displays as a comment or executes as a command or program.

Obviously problems can occur, just like they can if you enter each

Typesetting Conventions

We use `THIS typeface` to show exactly what you should type at your keyboard, with the special symbols ^ for the Control key (CTRL, CNTL or ALT on some keyboards), and ↵ for the Return key (RET, CR or Carriage Return on some keyboards). We also use `This Typeface` to show messages displayed by the system or by programs.

```
: THIS IS A SAMPLE BATCH FILE CALLED SAMPLE.SUB
: LINES BEGINNING WITH A COLON (:) ARE REMARKS OR COMMENTS
: LINES NOT BEGINNING WITH A COLON ARE PROGRAMS OR COMMANDS
DIR
STAT
: THESE TWO COMMANDS PRINT A DIRECTORY, THEN CHECK THE
: AMOUNT OF SPACE LEFT ON THE DISK IN THE CURRENT DRIVE
```

Figure 1a. *In CP/M (CP/M-80, CP/M-86, MP/M, etc.), you create a submit file and use it with the SUBMIT utility. Comments are lines beginning with a colon (:) or semicolon (;).*

```
REM THIS IS A SAMPLE BATCH FILE
REM LINES BEGINNING WITH ``REM'' ARE REMARKS OR COMMENTS
REM LINES NOT BEGINNING WITH ``REM'' ARE PROGRAMS OR COMMANDS
DIR
CHKDSK
REM THESE TWO COMMANDS PRINT A DIRECTORY, THEN CHECK THE
REM AMOUNT OF SPACE LEFT ON THE DISK IN THE CURRENT DRIVE
```

Figure 1b. *In MS-DOS, you create a batch file that runs as a command. Comments are lines beginning with the word REM.*

command individually. Be careful with any command which destroys information, like ERA, DELETE, FORMAT, COPY or PIP. If your batch file contains such commands, test it thoroughly with sample files before using it with "live" information.

Creating a Batch File

I use two different word processing programs, Select® and WordStar®, to show how to create batch files. The basic concepts are the same for most other word processing programs.

My explanations assume you know a little about your computer, operating system and word processor. The most critical things you need to know are:

1. How to turn on the computer.
2. How to start up your word processor.
3. How to go into your word processor's "Alter," "Non-Document," or "Programming" mode of operation.
4. How to name a file in your word processor.
5. How to insert and delete words, and move the cursor in the word processor.
6. What commands copy and erase files.
7. How to save a file in your word processor.
8. How to exit from your word processor.

When you think about these items, these assumptions are very minor. To do most work with your computer, these minimal skills are needed.

To start, turn on your computer. Insert the disks or other procedures necessary to load the operating system, whether CP/M or MS-DOS. Do the same for your word processor. If you use Select, type `SELECT` and then press the Return key (shown as ⏎ in examples). If you use WordStar, type `WS`⏎ (or whatever name you use for your copy of WordStar).

The word processor's Main Menu appears. In Select, press `A` to enter the ALTER mode of operation (no Return necessary). An asterisk (*) appears in the upper right corner of the screen to indicate it is in ALTER mode. In WordStar, press `N` (no Return needed) to enter the non-document mode of operation. If you are using another word processor, make the choice which does not preset margins or other print-formatting features (i.e., the mode that uses no printing controls or print-time paragraph formatting).

In Select, press `C` from the Main Menu to create a new file. WordStar requests the file name immediately after pressing `N`. Type in the following file name:

CP/M

```
WPBACKUP.SUB⏎
```

MS-DOS

```
WPBACKUP.BAT⏎
```

Notice there is a difference between file names. CP/M requires that "batch files" (called *submit files* in CP/M) have the ".SUB" extension. Submit files are executed by the CP/M utility program called SUBMIT.

MS-DOS, on the other hand, requires that batch files have the ".BAT" extension, because this special file extension tells the operating system it is a BATch file. You don't use a special program to execute a batch file under MS-DOS — you simply type its name as a regular command.

To execute the CP/M submit file, you type:

```
;*    *    *    *    *    *    *    *    *    *    *    *    *    *    *
;*                    W O R D   P R O C E S S I N G                    *
;*                                                                     *
;*A U T O M A T E D   B A C K U P   P R O G R A M                      *
;*                                                                     *
;*                                                                     *
;*                            COPIES FILES                             *
;*                         FROM THE HARD DISK                          *
;*                       TO A BACKUP FLOPPY DISK                       *
;*                                                                     *
;*                      (Kaypro 10 CP/M Version)                       *
;*                            (WPBACKUP.SUB)                           *
;*    *    *    *    *    *    *    *    *    *    *    *    *    *    *
;
; This batch file BACKS UP word processing files from User 1
; of Drive A (hard disk) to User 0 of Drive C (floppy drive).
;
```

Word processing file backup is only one of several common uses for batch files. Another critical use is keeping backup copies of accounting data.

```
A>SUBMIT WPBACKUP↵
```

To execute the MS-DOS batch file, you type:

```
A>WPBACKUP↵
```

Rules Governing Batch Files

CP/M and MS-DOS batch files are similar. Both let you use comments freely throughout the batch file, and both let you execute most programs and commands without running into limitations.

MS-DOS, though, offers more batch commands than CP/M. These simple programming commands make MS-DOS batch files more powerful and flexible than CP/M batch files. The batch commands can be very useful, but you should not try to create complex batch files (like menus for executing application programs). While this can be done, there are more powerful products available from independent software vendors which are easier to use than a batch file menu.

CP/M batch rules are very simple. A colon (or semicolon) is the only batch command available. When put in column 1 it tells the SUBMIT program the rest of the line is a comment. If the colon is not in column 1, SUBMIT assumes the line is a command or program to execute. If it isn't, you get an error message. A program or command to be executed may begin in any column, but it is best to always put it in column 1.

In sum, CP/M batch rules are (1) start all comments, programs and commands in column 1, and (2) put a colon or semicolon in column 1 to indicate the line is a comment to be displayed on the screen.

MS-DOS batch rules are similar, but there are more commands. They include ECHO, FOR, GOTO, IF, SHIFT, PAUSE and REM. These

```
; The following word processing files will be backed up:
; =======> L??, R??, ENV, BK1 and ART
;
; Press <RETURN> to continue — Press <ctrl-C> to start over.
;
READY
;
;
;
; *****************************************************************
;
; Make sure the following steps are completed before continuing:
;
; 1. Get the correct back up floppy (safety, security or archive).
; Double check the label on the disk.
;
; 2. Place it in the floppy drive with the label facing up.
;
; 3. Close the floppy drive door.
;
; *****************************************************************
;
; Press <RETURN> to continue.
;
READY
;
;
; GETTING RID OF UNNECESSARY FILES
;
; ====> ERASE .BAK FILES
;
ERA A:*.BAK
;
;
; GETTING RID OF PRINT FILES (Select WP only)
;
; ====> ERASE .SEL FILES
;
ERA A:*.SEL
;
;
; READYING FLOPPY DISK FOR COPYING
;
C:
USER 0
A:DSKRESET
;
COPYING FILES FROM USER 1 OF DRIVE A
;
PIP C:=A:*.L??[V]
PIP C:=A:*.R??[V]
PIP C:=A:*.ENV[V]
PIP C:=A:*.B??[V]
PIP C:=A:*.ART[V]
;
;
READY
;
; **************************************************************
;
; To finish up, do the following:
;
; 1. Remove back up floppy from the floppy drive.
;
; 2. Put back up floppy away in a safe place.
;
; **************************************************************
;
;
READY
;
;
; ======> BACK UP COMPLETED <======
```

Figure 2a. *This CP/M batch submit file copies word processing files from user area 1 of drive A to user area 0 of drive C.*

```
REM *     *     *     *     *     *     *     *     *     *     *     *     *     *
REM *                         W O R D   P R O C E S S I N G                       *
REM *                                                                             *
REM *                 A U T O M A T E D   B A C K U P   P R O G R A M             *
REM *                                                                             *
REM *                                                                             *
REM *                                COPIES FILES                                 *
REM *                             FROM THE HARD DISK                              *
REM *                           TO A BACKUP FLOPPY DISK                           *
REM *                                                                             *
REM *                              (MS-DOS Version)                               *
REM *                               (WPBACKUP.BAT)                                *
REM *     *     *     *     *     *     *     *     *     *     *     *     *     *
REM
REM This batch file BACKS UP word processing files from
REM Drive A (floppy drive) to Drive B (floppy drive).
REM
REM The following word processing files will be backed up:
REM =======> L??, R??, ENV, BK1 and ART
REM
REM Press <RETURN> to continue — Press <ctrl-BREAK> to start over.
REM
PAUSE
REM
REM
REM
REM ***************************************************************
REM
REM Make sure the following steps are completed before continuing:
REM
REM 1. Get the correct back up floppy (safety, security or archive).
REM Double check the label on the disk.
```

commands make it possible to determine whether to display a comment or not, whether to repeat a series of instructions, and whether to stop execution until some action is taken.

MS-DOS batch rules also require the first letter of a program, command or batch command start in column 1. MS-DOS reads the characters until it sees a space. It then determines whether these characters are a program, command or a batch command, then reacts accordingly.

For example, if the first three characters are `REM`, this tells MS-DOS the rest of the line is a remark or comment. If the characters are `ECHO`, it looks to the next few characters (until it sees a space) to determine whether you want comments and commands displayed (`ON`) or not (`OFF`).

Of the remaining batch commands, PAUSE is the most useful to the average batch file user. It stops execution of the batch file until a key is pressed. This temporary halt is very useful when you need to change disks, or if instructions must be read from the screen before continuing the operation.

Finally, both CP/M and MS-DOS allow you to send information to the batch file before it starts. Such information passes to the batch file in the form of *parameters*. Up to ten parameters may be identified when you type the command to execute either a CP/M or MS-DOS batch file. MS-DOS allows more parameters when you use the SHIFT command, but the casual user will rarely need more than ten parameters.

Word Processing Files: A Common Example

Let's begin with a common example — a file backup operation of documents, reports and other word processing text files. Sample batch files for CP/M and MS-DOS are shown in figures 2a and 2b.

To understand what these batch files do, and to understand some of the other procedures you must develop, several items must be explained.

Word processing files must have consistent identification. In this example, all letters and memos are identified by the filename extension ".L??". The ".ENV" extension refers to the file holding a name and address in an envelope format. The name of the file should be the same as the letter it corresponds to.

The extension ".R??" refers to reports or analyses. The ".B??" and ".ART" extensions refer to chapters of a book and an article, respectively. The "??" again identifies for whom the document was typed. Any 8 character name may identify the content of the letter or memo.

All backup disks must be properly labelled. I briefly refer to different backup disks — Safety, Security and Archive. Unless these are properly labelled, you'll have a hard time finding the most recent revision of a document.

Disks must be safely stored away. Put them in their sleeves. Don't paperclip anything to them. Put the disks in a sturdy cardboard or plastic box, and put it away from potentially harmful sources like the sun, a heater and children. Remember, this is important information, treat it that way.

Finally, if you do any serious word processing or accounting work, you should keep multiple generations of files. For this word processing example, I refer to Safety, Security and Archive backup copies.

Safety Backup is a disk you copy files to periodically during a day. It safeguards against your Work Disk (whether floppy or hard disk) from such common maladies as static electricity. Security Backup is a weekly backup of all text files. The final generation of backup is Archive. It is created from the Work Disk and Security copies. It consists of text files you decide you want to keep for a long time. Usually Archive Disks are made monthly.

Figures 2a and 2b show the CP/M and MS-DOS batch files for the backing up operations. Many of the lines are comments which are displayed as the batch file executes. Comments are used here to include explanations and instructions as well as white space,

```
REM
REM 2. Place it in the Drive B with the label facing up.
REM
REM 3. Get the correct word processing Work Disk.
REM Double check the label on the disk.
REM
REM 4. Close the both floppy drive doors.
REM
REM **************************************************************
REM
REM
REM
REM
REM
PAUSE
REM
REM GETTING RID OF UNNECESSARY FILES
REM
REM ====) DELETE .BAK FILES
REM
DEL A:*.BAK
REM
REM ====) DELETE .SEL FILES (Select WP only)
REM
DEL A:*.SEL
REM
REM
REM COPYING FILES FROM DRIVE A
REM
COPY A:*.L?? B:
COPY A:*.R?? B:
COPY A:*.ENV B:
COPY A:*.B?? B:
COPY A:*.ART B:
REM
REM
PAUSE
REM
REM **************************************************************
REM
REM To finish up, do the following:
REM
REM 1. Remove back up floppy from Drive B.
REM
REM 2. Put back up floppy away in a safe place.
REM
REM 3. Remove Work Disk from Drive A.
REM
REM **************************************************************
REM
REM
PAUSE
REM
REM
REM ======) BACK UP COMPLETED (======
```

Figure 2b. *This MS-DOS batch file copies word processing files from drive A to drive B.*

```
;*    *    *    *    *    *    *    *    *    *    *    *    *    ****
;*                  C L I E N T   D A T A   F I L E S               *
;*                                                                  *
;*            A U T O M A T E D   B A C K U P   P R O G R A M       *
;*                                                                  *
;*                                                                  *
;*                      COPIES CLIENT DATA FILES                    *
;*                        FROM THE HARD DISK                        *
;*                       TO A BACKUP FLOPPY DISK                    *
;*                                                                  *
;*                           (CP/M Version)                         *
;*                           (CFBACKUP.SUB)                         *
;*    *    *    *    *    *    *    *    *    *    *    *    *    ****
;
; This batch command file BACKS UP your T/PAC client files from
; User 1 of Drive B, C and D to User 0 of Drive F (floppy drive).
;
; =====> The client to be backed up is: $1
;
; If this is correct, press <RETURN>.
; If this NOT correct, press <ctrl-C> to start over.
;
READY
;
;
;
; *****************************************************************
;
; Make sure the following steps are completed before continuing:
;
; 1. Get the backup floppy for this client.
; Double check the label on the disk.
;
; 2. Place it in the floppy drive.
;
; 3. Close the floppy drive door.
;
; *****************************************************************
;
;
;
;
;
; Press <RETURN> to continue.
;
READY
;
; READYING FLOPPY DISK FOR COPYING
;
F:
USER 0
A:DSKRESET
;
;
; COPYING FILES FROM USER 1 OF DRIVE B.
;
PIP F:[G0]=B:???$1.DAT[G1V]
```

boxes and arrows for easier reading and use. I strongly recommend you do this for any batch files other people will use or which you only occasionally use.

As you can see in figures 2a and 2b, the two batch files are nearly identical. Both have the same boxed heading and general description of purpose. They tell the operator what is being backed up and from where and to where the files are copied.

Also, the operator is given the opportunity to stop the batch file by pressing Control-C or Control-Break at the key point. This may be necessary if the wrong batch file is started, or the operator forgot some preliminary tasks like formatting disks.

One other difference is the type of disk being copied. The CP/M batch submit file (figure 2a) copies a hard disk to a floppy. The MS-DOS batch file copies one floppy disk to another floppy. Either file can easily be changed for your system.

The CP/M batch submit file uses a special program to perform what MS-DOS's PAUSE batch command does. PAUSE tells the batch file to stop executing instructions until a key is pressed. The READY program does the same thing as PAUSE. READY is a compiled program which simply tells you to press your Return key to continue. It does nothing else.

READY is part of the installation programs and submit files provided with the Select Word Processor. If you have Select, use READY to pause executing instructions. If you don't have Select and don't have a compiled programming language, consult a friend who programs or a computer store to have one developed. It should take an average programmer less than fifteen minutes to create.

The next group of comments simply tell the operator to prepare for copying. They are designed for a novice user, and can be expanded or contracted as you wish.

You may have noticed the first two groups of lines are both 23 lines or less. Since these two groups have instructions and explanation which should be read, they are designed to fit on a single 24 line screen. Only what is important at that time appears on the screen. This design can be very useful when creating batch files, to be used by novice operators.

The next group of lines is the meat of the batch files. They contain the commands to erase unnecessary files. Both Select and WordStar create files with the ".BAK" extension whenever you edit an existing text file. It is exactly the same size as the original. While the ".BAK" file could be used if you inadvertently erased the original, the Safety Backup file created by using the batch file is better.

The batch file safeguards against both accidental deletion of a text file (like ".BAK") as well as damage to the hard or floppy work disk. In addition, Select creates an ".SEL" file when you print a document. Again, it could be useful, but in two years of using Select I've never needed an ".SEL" file. I prefer to erase them from my disk, but you should determine this for yourself.

The CP/M and MS-DOS commands to erase and copy files are similar. CP/M uses ERA and PIP to perform these tasks, while MS-DOS uses DEL and COPY. When erasing files under either operating system I strongly suggest putting the drive letter before the filename. You are more likely to type the right drive letter so that important files are not accidentally destroyed.

CP/M and MS-DOS differ significantly in the next group of lines. Under CP/M you must "reset" the floppy drive after changing a disk since the last "warm boot". (Briefly, "warm boot" means to reload essential parts of the operating system, and "reset" is the effect of a warm boot.) A Control-C performs a warm boot of the entire computer. It throws any programs in memory out of memory: the batch file is an example (look for Control-C in the first group of lines).

CP/M's SUBMIT program shows its limitations here. SUBMIT doesn't accept a Control character like Control-C. (Several manuals say there is a way, but I've yet to succeed in using Control characters on two different CP/M computers.) Also, no manual indicates how to reset just the drive and

```
;
;
; COPYING FILES FROM USER 1 OF DRIVE C.
;
PIP F:[GO]=C:???$1.DAT[G1V]
;
;
; COPYING PAYROLL, IF ANY EXIST.
;
PIP F:[GO]=$2:???$1.DAT[G1V]
;
;
; Press <RETURN> to continue.
READY
;
;
; ****************************************************************
;
; To finish up, do the following:
;
; 1. Remove floppy drive. Make sure it has a correct label.
;
; 2. Put backup floppy away in proper place.
;
; ****************************************************************
;
;
READY
;
;
; ======> BACKUP FOR $1 COMPLETED <======
```

Figure 3a. *This CP/M batch submit file copies T/PAC client files from user area 1 of drive B, C and D to user area 0 of drive F.*

```
REM *     *     *     *     *     *     *     *     *     *     *     *     *     ****
REM *                      C L I E N T   D A T A   F I L E S                        *
REM *                                                                               *
REM *                 A U T O M A T E D   B A C K U P   P R O G R A M               *
REM *                                                                               *
REM *                                                                               *
REM *                          COPIES CLIENT DATA FILES                             *
REM *                             FROM THE HARD DISK                                *
REM *                          TO A BACKUP FLOPPY DISK                              *
REM *                                                                               *
REM *                              (MS-DOS Version)                                 *
REM *                               (CFBACKUP.BAT)                                  *
REM *     *     *     *     *     *     *     *     *     *     *     *     *     ****
REM
REM This batch command file BACKS UP your T/PAC client files from
REM Drive B, C and D to Drive F (floppy drive).
REM
REM =====> The client to be backed up is: $1
REM
REM If this is correct, press <RETURN>.
REM If this NOT correct, press <ctrl-BREAK> to start over.
REM
PAUSE
```

```
REM
REM
REM
REM *******************************************************************
REM
REM Make sure the following steps are completed before continuing:
REM
REM 1. Get the backup floppy for this client.
REM Double check the label on the disk.
REM
REM 2. Place it in the floppy drive.
REM
REM 3. Close the floppy drive door.
REM
REM *******************************************************************
REM
REM
REM
REM
REM
REM
PAUSE
REM
REM
REM COPYING FILES FROM DRIVE B.
REM
COPY B:???$1.DAT F:
REM
REM COPYING FILES FROM DRIVE C.
REM
COPY C:???$1.DAT F:
REM
REM COPYING PAYROLL, IF ANY EXIST.
REM
COPY $2:???$1.DAT F:
REM
REM
PAUSE
REM
REM
REM *******************************************************************
REM
REM To finish up, do the following:
REM
REM 1. Remove floppy drive. Make sure it has a correct label.
REM
REM 2. Put backup floppy away in its proper place.
REM
REM *******************************************************************
REM
REM
PAUSE
REM
REM
REM ======) BACK UP FOR $1 COMPLETED (======
```

Figure 3b. *This MS-DOS batch file copies T/PAC client files from user area 1 of drive B, C and D to user area 0 of drive F.*

not the whole computer system.

My solution was to use a simple program again (see following note). I refer to one provided with MP/M (a multi-user version of CP/M) called DSKRESET. It resets the floppy drive without warm booting the whole computer system. A friend who programs or a computer store should be consulted for a program like DSKRESET. Again, less than an hour is needed to produce the program.

NOTE

You may have problems getting a program like DSKRESET. While this is the most elegant method and easiest for novice operators, another technique is available. Before you start the batch file, insert the backup floppy disk in the drive (drive C if you have a computer system like the Kaypro 10 version example above, or drive B if you have a dual floppy system). Now press Control-C by holding down the Control key while pressing the letter C. This combination performs a warm boot on the computer system, and the floppy drive is reset so you can write to it. *Do not* open the floppy drive (drive C or drive B), because this triggers the operating system and you'll have to reboot again. This method eliminates the need for the second group of lines, so you should delete them from the batch file.

Finally, the last group of lines prompt the operator to remove the floppy disks and store them away safely. The back up of documents is complete.

Advanced Uses

Word processing file backup is only one of several common uses for batch files. Another critical use is keeping backup copies of accounting data. It isn't just good business practice, it is required by the IRS. They see business records as business records, whether they're on paper or on a disk. The IRS require they be kept for at least 5 years. Thus, backup of accounting data is very important.

Figures 3a and 3b are examples of a batch file used in a CPA office to back up client records. The CP/M version is called CFBACKUP.SUB and the MS-DOS is called CFBACKUP.BAT. The point of this example is to show how information can be passed to the batch file before it begins executing instructions.

As you can see the batch files are very similar in form and content. In fact, they were designed to substantially differ only in the key wording and the commands which execute the copying. This similarity makes it easier for novice users to work with the batch file. It also makes it easier to create a new batch file.

This example deals with the accounting data of multiple businesses maintained by a CPA. Because multiple clients are involved, the software keeps track of the clients with an ID code. During back up the client's accounting records are identified using this code; in fact, the code is part of the file name.

In the first group of lines you see `$1`. This is a *parameter*. It represents the first piece of information sent to the batch file before execution. This same parameter is used in the third group of lines, which is where the commands to copy the files are. Here it is inserted into the file name to identify the file to be copied to the back up disk.

Notice the `$2` in the `COPYING PAYROLL...` group of lines. This second parameter is used in a novel way. It tells the batch file whether payroll records exist for the client being copied.

It is novel because the client may not have payroll records. If you attempt to copy a file which doesn't exist, the batch file stops running and an error message is given. Two batch files, one for clients with payroll and another for those who don't, appear to be necessary.

There is a way around this extra work and potentially confusing procedure. The second parameter tells the batch file on which drive the payroll files are stored. If the client has payroll, the actual drive is given. If the client *does not* have payroll, give *another drive* which has client records.

For example, assume the client's accounting records are stored on drives B, C and D. Drive D contains all client's payroll records. To start the client backup file, type one of the following:

(1) `SUBMIT CFBACKUP BW123 D` ↵
(CP/M version with payroll.)

(2) `SUBMIT CFBACKUP BW123 B` ↵
(CP/M version without payroll.)

(3) `CFBACKUP BW123 D` ↵
(MS-DOS version with payroll.)

(4) `CFBACKUP BW123 B` ↵
(MS-DOS version without payroll.)

If you type example (1) or (3) above, `BW123` is parameter one (`$1` under CP/M) and `D` is parameter two (`$2` under CP/M). It tells the batch files to copy client BW123 from drive B, C and D (parameter 2). It you type (2) or (4) above, `BW123` is parameter one (`%1` under MS-DOS) and and `B` is parameter two (`%2` under MS-DOS). Remember, all payroll files are on drive D. So if I tell it to copy files off drive B, it just copies the client files again.

Thus, by telling it what drive to get the "payroll" files from, I can indirectly tell the batch file whether I have payroll files.

The possibilities for batch file operations are as endless as the applications for personal computers. Nearly every computer application requires file backup operations, and most require repetitive commands that can be performed by batch files without as many errors that could happen when you type each command. The basic examples shown here should be a starting point for your applications.

Knowledge And Mind Amplification System.

A unique blend of familiar productivity tools with dynamic new ones:
- ☐ **Outline Processing** lets you organize your ideas in a familiar outline form.
- ☐ **Information Retrieval** keeps your thoughts at your fingertips with astonishing speed.
- ☐ **Word Processing** fills out the structure of your ideas with text.
- ☐ **Telecommunications** lets you set up a structured, electronic bulletin board.
- ☐ **Programming Environment** lets you get under the hood and extend your horizons.

It all adds up to **awesome Knowledge Processing power.** Your chance to join the next wave of the integrated software revolution.

Available for CP/M, Z80 systems. Special introductory offer: $147. Send now for your **free** copy of *The KAMAS Report.*

COMPUSOPHIC SYSTEMS
Dept. 131 • 2525 SW 224th Ave.
Aloha, Oregon 97006 • (503) 649-3765

KAMAS is a trademark of Compusophic Systems. CP/M is a registered trademark of Digital Research, Inc. Z80 is a registered trademark of Zilog, Inc.

Circle number 6 on the reader service card.

Thanks

This is a letter of thanks for all the hard work you have done to produce a CP/M and IBM users' magazine that contains so many lucid and in-depth articles. This will make my life much easier in teaching the visually impaired and the blind to use the Enable Reader Program when I complete it under CP/M. For over a year I have worked with the blind community in Berkeley to develop a program that meets their needs and follows a strict set of design goals.

Specifically, I would like to thank Steve Rosenthal for his articles on CP/M Plus (*User's Guide* #4 and #5). Although the children who are using the program do not need it now, the college students and graduates who are using it have already commented on how the documentation inside your magazine is better than any they have had to struggle with in manuals and software guides. Keep it coming and I hope to see some articles on how the disabled community is networking with CP/M.

Robert Artusy
Berkeley, CA.

The introductory articles to CP/M contained in number 1 are great. I got a quick look at someone else's copy, and now I want one of my own.

It is hard to find good, easy-to-read introductions with lots of examples. Most software reviewers in magazines indicate their likes and dislikes according to their tastes, but do not give enough examples for the reader to make a personal judgement. Your articles take the reader through some valuable examples, which allow the flavor of using that software to escape any bias of the author. Keep up the good work!

James White
Norman, OK

The main problem with your magazine is that if it is left around, it is sure to be snatched up!
Richard Healey
Washington, DC

IBM PC Coverage

I also know how it feels to shoot oneself in the foot. Fortunately for me and scores of others you handled the IBM PC situation in a manner consistent with what I have come to expect from *User's Guide.* No doubt I will remain a faithful reader for many months to come.

One area of my interest that I'm sure is shared by many others is: what is the best way to upgrade to 16-bit processing power without wasting my present hardware investment? A few companies have introduced add-on boards that will supposedly allow the 8-bit Z-80 based system to run MS-DOS and in some cases even be PC-compatible.

I think it would be an excellent investment of time to review some of these devices.

Bob Blum
Norcross, Georgia

Right you are, Bob. We intend to step up our coverage of MS-DOS with evaluations of add-ons for the Kaypro, some S-100 boards, add-ons for other Z80-based computers, and computers that offer both 8-bit and 16-bit environments. We also intend to cover the data file conversion utilities.

I continue to find *User's Guide* to be a very informative magazine. I am not upset at your addition of MS/DOS to your coverage, as I, like many others, now work with a PC compatible at work (an Anderson-Jacobson Passport, nee Hyperion), while my home system is (and will be for quite awhile into the future) a CP/M machine (an Osborne 1).

Incidently, I was pleasantly surprised to see that *User's Guide* seems to have a significant role in the return of the Osborne Computer magazine *Portable Companion.* Thanks again for providing many hours of educational and enjoyable reading!

Curtis P. Hedman
Saint Paul, MN

Request for help for Sanyo owner (reprinted from issue #7)

You say MODEM7 is "free", if you know where to find it, and you list RCPM systems (phone numbers only) that presumably can unload it to me on request. But if I were able to download from an RCPM, I wouldn't need MODEM7! When I ask about it, dealers either don't know or won't talk about it. They don't know of COMMXPAC, either, and I know of no local communications program user group nor even how to go about finding one.

How and where can I, lacking a user group or the ability to go online, get a "free" copy of MODEM7? What commercial programs besides COMMX and MITE offer the MODEM7 protocol?

Thank you - for this and for *User's Guide*, a readable, refreshing oasis of much-needed CP/M help.

George K. Johnson
St. Louis, MO

The list of RCP/M systems will be published again in User's Guide *#11, updated. The list contains no addresses by request of the SYSOPS (system operators), who provide help on-line, but do not want to consult by mail for free. They provide a valuable service already, which your dealer should take advantage of.*

Your dealer can and probably should make MODEM7 available free with every system he sells. Many dealers do this and say it is worth it to

Continued on page 64.

FOG

The **FIRST OSBORNE GROUP (FOG)** has been formed as a User's Group for persons using or interested in portable computers and/or the CP/M disk operating system with related software. In May of 1984, there was nearly 12,000 members from around the world. Most attend local group meetings at over 300 locations.

Computer systems owned or used by members include the Osborne 1 (single and double density), the Osborne Executive 1, all models of the Morrow MicroDecision, the Zorba, all models of the KayPro, several MicroMates, and many more. Special interest groups organized to augment a network of local group meetings include dBase II, Ham radio operators, Personal Pearl.

FOG was started in October of 1981 by a small band of early buyers of the Osborne 1. The primary purpose was to organize a library of public domain software to run on the Osborne 1. A newsletter was quickly started to act as a focal point for the group's activities. The large number of excellent contributions to both the library and the newsletter has produced a library of 200 disks (as of April, 1984) and a nicely typeset (56 or more pages) monthly publication. All back issues of the ***FOGHORN*** are available for a nominal fee which includes shipping in the U.S. Contributions are currently being solicited for bulletin board systems in addition to the three currently in operation. All systems accept both 300 baud and 1200 baud. Phone numbers for existing systems are:

System #1 — Daly City, CA	(415) 755-2030
System #2 — Vancouver, BC	(604) 596-0314
System #3 — Daly City, CA	(415) 992-8542

While the meetings are organized on a local basis, over 100 of these local groups have joined the FOG network, thus increasing the sharing of information, tips, problems and so on. Those local groups which opt to formally join the FOG network receive a portion of local member dues to assist with the cost of maintaining a local copy of the disk library.

The FOG library is currently maintained on the Osborne 1 single density format but separate libraries are being established for the other computer formats. The library files are carefully screened and divided into category types (utilities, games, applications, and computer languages are the four major categories). Programs which contain run or other errors are put into the hacker section so interested members can fix them and resubmit for inclusion in the correct section. Items which do not fit into one of these categories are in the miscellaneous section. A catalog and descriptions of all the discs is maintained in the library section.

Dues in FOG are $24.00 per year. This entitles each member to a copy of the ***FOGHORN*** each month as well as access to the disk library. Local group meetings are open to the public without charge although most restrict access to the library to the membership. The FOG library contains only public domain software. Piracy (the copying of proprietary software) is strongly condemned.

In the United States, the ***FOGHORN*** is normally mailed by non-profit bulk mail. (FOG is a corporation in the state of California and has obtained its non-profit, tax exempt status from both the state and federal governments.) For those members who live out of the country or who prefer first class delivery of their ***FOGHORN***, additional postage must be added to the annual dues. See the chart below for details.

If you are interested in joining a self-help organization to increase you knowledge and the use of your computer, use the application below (or a copy of it). Generally, memberships received at the FOG office prior to the 15th of the month are entered in time to receive the next month's ***FOGHORN***.

If you know of a local group which might be interested in joining the FOG network, please send all details (meeting dates and places, officers, and how interested local computer owners can join). We will send you an information packet on becoming an Affiliated Member Organization.

For your records, the address of FOG is P. O. Box 3474, Daly City, CA, 94015-0474. Please allow at least two months for the arrival of your first ***FOGHORN*** since bulk mail can take as much as nine weeks. (The post office says that it should only take about three week for non-profit bulk mail but some members on the East Coast have experienced longer delays.) A membership card will be processed within a week of the receipt of your dues.

ADDITIONAL POSTAL CHARGE CHART

Please add the appropriate amount to your dues payment.

Canada & Mexico (First Class Airmail delivery)	ADD...$ 6.00
Members with U.S. addresses who prefer First Class delivery to bulk mail	ADD...$ 6.00
Central & South America, Caribbean, & Europe (Airmail First Class delivery)	ADD...$12.00
Asia, Africa, & Far East (Airmail First Class)	ADD...$15.00
Out of North America preferring surface mail — delivery not guaranteed	ADD...$ 6.00

CUT HERE

NAME: ______

Company (if part of mailing address): ______

ADDRESS: ______

CITY: ______

STATE: ______ ZIP or MAIL CODE: ______ COUNTRY: ______

HOME & WORK PHONES: ______

MEMBER OF WHICH LOCAL GROUP? ______

COMPUTER TYPE? ______

MODEM TYPE? ______ PRINTER TYPE? ______

INTERESTS? ______

Send this completed application **AND** your payment to:

FIRST **O**SBORNE **G**ROUP
P. O. Box 3474
Daly City, CA 94015-0474
United States of America
Phone: (415) 755-4140

Be sure to include any required postage surcharge.
Your membership card will be sent by 1st class mail.

Circle number 9 on the reader service card.

Letters *Continued from page 62.*

them, because the happy customer has less after-sale questions, and will buy additional software, etc. from the dealer he trusts.

The public domain (free) software does not compete with any commercial products your dealer carries. Most computer users use both commercial and public domain software.

You can order COMMX directly from Hawkeye Grafix, 23914 Mobile, Canoga Park, CA 91307 for $195 for the CP/M micro version.

If you know of a local computer user group, you will likely find that at least one member knows about communications software. Even if they aren't familiar with your computer they can be of help. Support or start your own local user group, or join a national one.

Modem 7 Research

I just received my copy of *User's Guide* #7 and noticed the letter from George Johnson. This past weekend I solved the same problem he writes about.

To get a copy of Modem7 I had to buy a commercial software package, Move-It, to operate my modem. I then "captured" a very short program from a local RCP/M called BMODEM.BAS which runs under Microsoft Basic-80 and allows one to download programs from a RCP/M. I then called a local RCP/M, invoked XMODEM, exited from Move-It, loaded MBASIC and BMODEM, and proceeded to successfully download USQ, MDM730 and the overlay file for my computer, plus the operating instructions. Modifying the overlay for my computer was a matter of minutes using WordStar, ASM and DDT. It worked the first time.

Your upcoming article in *User's Guide* might mention this, plus a listing of the overlays available. I just happened to know which overlay was written for my computer. But if I had not, it would have been very difficult.

One more suggestion: how about a listing of the most popular RCP/M programs and their use/purpose? The article in *User's Guide* #3 by Smith and Bove listed several, but the number available is simply staggering!

Your magazine is great! Keep up the good work and I hope that you good folks make it in the marketplace of computer magazines.

Hugh T. Hodges
Glendale, CA

High on our list of most requested articles is a guide to the best of the public domain software. We are working on it, and need all the input you can give. Write to: Modem7 Research, PO Box 5245, Redwood City, CA 94063.

Tutorials Suggested

How about an article on the BMC-IF 80020 computer?

Sue Heron
Arlington Hts, IL

Sometime in the future, would you consider an evaluation of the DataFlex data base program? I am told that it is better than dBASE II, and I would like to see your evaluation and opinion. Keep up the good work.

Lillian Huber
President of Alpha Enterprises
Alpharetta, GA

We want to hear your suggestions of products you are seriously interested in. Our reputation for independent reviews has resulted in a flood of packages to review. Some of them have expensive price tags and packaging, others are very inexpensive. Although the packages and prices vary, the performance of the products is impossible to gauge from the packaging.

I have an Eagle 2E computer and I am interested in what CP/M software is available for it.

R. Hersted
Sacramento, CA

Support should be one of the factors you consider before you buy your computer, unless you have unlimited time to spend teaching yourself. You may want to hire a consultant to help you if there is no user group nearby.

If you have a modem and communications software, you can call RCP/Ms (remote CP/M systems) and download (receive) public domain (free) software. You can call information services such as The Source and CompuServe and join their CP/M SIGs. You can call BBSs (bulletin board systems - similar to RCP/Ms) and leave messages asking for help. Your best sources of information are other users, whether you meet in person at user group meetings, or carry on a teleconference by computer.

Bad Load

I have enjoyed your magazine and recommend it as the only literate magazine publication in the field. You have the knack of putting computer jargon in understandable terms.

W. A. Fauve, Sr. of Baton Rouge, LA may be having problems with a CP/M system that is not configured for enough memory. I had the same problem with Heathkit's first venture into the CP/M operating system. Since then their documentation has become superb.

Their first documentation was that of Digital Research: complete gibberish.

The system came configured to 32K and it was up to the user to make the system recognize all of the memory and peripherals that were present; including the kinds of disk drives that were added later.

Not even Microsoft Basic would load. Attempts to load it led to a `BAD LOAD ERROR`.

Then I noticed that the signon message said that the system was configured for "32K of memory" when I had 64k installed. After much reading I discovered MOVCPM and was able to reconfigure the system to all of the memory that was present so that it could accept Microsoft Basic.

A defective memory cell, particularly in the buffers that are used for initial disk access, will give the same problem. That one can be harder to solve.

Hope that this will be helpful to W.

A. Fauve, Sr.
Keep up the good writing.

Obert W. Rasch
Johnson City, TN

The message `BAD LOAD` means the CCP discovered the size of the ".COM" file being loaded exceeds the available memory. Since PIP works, your CP/M is almost certainly functional. The most likely thing is that the application program that gets the diagnostic is simply too big to run on your computer, i.e., you need more memory. It is possible, however, that the ".COM" file that will not load was created by SAVE on a larger machine, and the image saved was larger than it needed to be. If that is the case, then the program may run even though CCP says `BAD LOAD`. To find out, create a zero length file with the command:

```
SAVE 0 TRYIT.COM↵
```

Then call your program. When CCP says `BAD LOAD`, execute the loaded program anyhow with the command:

```
TRYIT↵
```

It won't hurt to try it. Dismount all disks first, though, just in case it does.

Jack D.Dennon
Warrenton, OR.

Reprints

Would you please inform me what your policy or practice is on reprints. The Fielding Institute has adopted Compuserve as its EMail facility for the time being, and we find the article called Electronic Mail (which I have seen on Xerox) from *User's Guide* #6 to be vastly superior to Compuserve's manual.

If it is representative of the writing in your magazine, it is likely we would occasionally want reprints of other articles, too.

Hallock Hoffman
Executive vice president
The Fielding Institute
Santa Barbara, CA

The articles in User's Guide *are copyrighted. The copyright notice appears at the bottom of every page you xerox. It is illegal to xerox any copyrighted material without permission.*

We provide article reprints at cost to those who are willing to pay for them. If the order is small, we often find it more cost effective to sell the whole magazines containing the articles. We sell multiple copies of these issues at a discounted rate.

Please write to: Reprints, User's Guide, PO Box 5245, Redwood City, CA 94063, if you need multiple copies of an article for your school or business.

A POURNELLE USERS GUIDE

TWO NEW POURNELLE USERS GUIDES FROM BAEN COMPUTER BOOKS

The newest in a continuing series commissioned, selected, edited or approved by Jerry Pournelle— books you can depend on by a name you can trust!

THE ESSENTIAL USERS GUIDE TO THE IBM PC, XT, AND PCjr
by Dian Girard

Of *course* you want an IBM— but *which* IBM?

READ THIS BOOK

This complete and easy-to-understand guide to the IBM personal computer family includes an Appendix with extensive technical details plus programs that can be keyed in and are also available on disk!

NOVEMBER 55922-2 • 256 pp. • $6.95

THE USERS GUIDE TO CP/M SYSTEMS
Tony Bove and Cheryl Rhodes

Finally! An *understandable* training manual that offers hands-on instruction for the beginner. Newcomers to the world of personal computing need *friendly* help, not confusing instructions that require decoding before they can be applied. This handbook, written by the editors of *Users Guide to CP/M* magazine, was specifically designed for the novice who wants to dive right in and learn while doing. Provides in-depth instruction in two crucial areas: word processing and programming.

NOVEMBER 55921-4 • 224 pp. • $6.95

INCLUDES CP/M PLUS AND CP/M 86

Distributed by Simon & Schuster Mass Merchandise Sales Company
1230 Avenue of the Americas • New York, N.Y. 10020

Circle number 2 on the reader service card.

Computer Posters

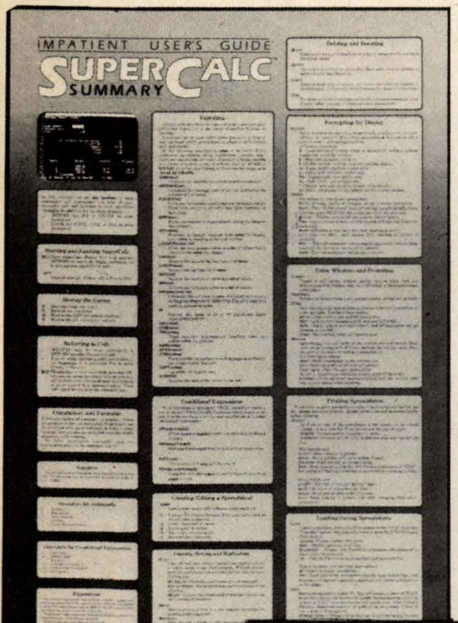
Computer Wimp is the poster based on the best-selling book — lists 166 things you should know about computers. Funny and informative. ***Murphy's Computer Law***. Murphy and computers were made for each other. Our Murphy's Law poster has sold over 750,000 copies.

Flatware™ is our name for a line of full size posters designed as a reference tool for popular software now in wide use in homes and offices. If placed on the wall near the computer screen, these posters are an easy-to-read guide.

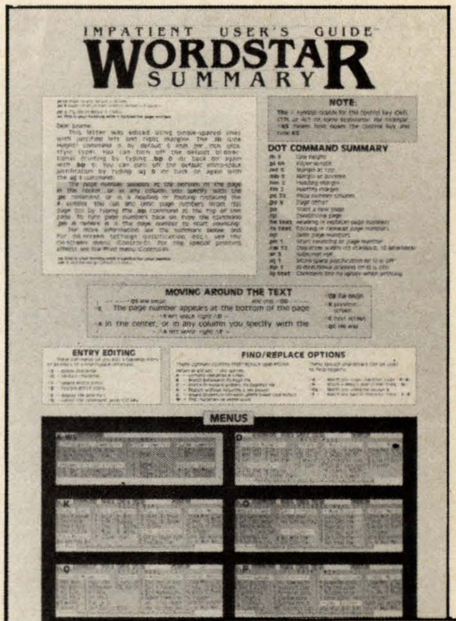
WordStar® presents the menu screens and control characters for one of the most widely used word processing software. ***SuperCalc®*** summarizes the commands needed to operate the popular financial spreadsheet software.

Celestial Arts / Ten Speed Press

P. O. BOX 7327
BERKELEY, CA 94707
(415) 524-1801


IMPATIENT USER'S GUIDE
SUPERCALC
SUMMARY

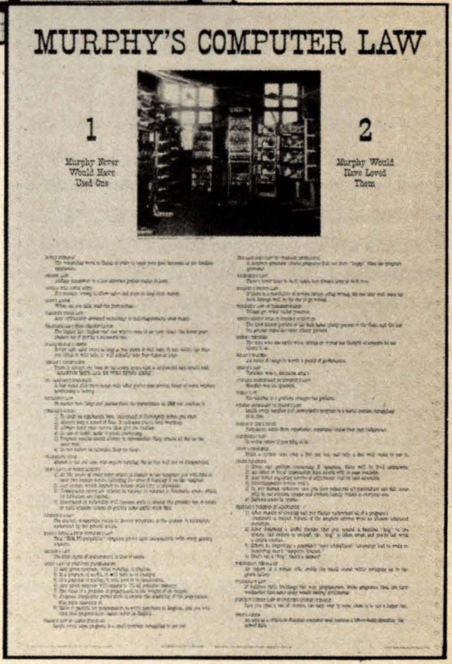
FW-1 Supercalc, $5.00


IMPATIENT USER'S GUIDE
WORDSTAR
SUMMARY
NOTE
DOT COMMAND SUMMARY
MOVING AROUND THE TEXT
ENTRY EDITING
FIND/REPLACE OPTIONS
MENUS

FW-2
Wordstar, $5.00

COMPUTER WIMP
166 Things you should know before you buy a COMPUTER!
—or before you give up on the one you already have.

TSP-1 Computer Wimp, $3.50


MURPHY'S COMPUTER LAW
1
Murphy Never Would Have Used One
2
Murphy Would Have Loved Them

SP-155 Computer Murphy's, $3.50

Back Issues of The Portable Companion!

Back issues at $4 each ($5.50 foreign):

Issue# ____________ Qty. ________ at $4 each ($5.50 foreign) = $____________

August/September 1982:
Reach Out and Touch Another Computer • Why Can't Computers Be Fun • The Colorful Strobe Plotter • Inside Your Osborne 1 • Draw Cards Using MBASIC™ • Harvard Bans the Osborne 1 • Inside/Outside Osborne 1 • Mailing Labels with dBASE II™ • Monkey Puzzler Returns •

October/November 1982:
Automatic Line Numbering • I/O, I/O, It's Off to Work We Go • Where Did All the Memory Go? • Two-and-a-Half Utilities for dBASE II • Random Tracks • Connecting the Diablo 630 Printer • A Primer on Hex and ASCII •

February/March 1983:
Dragons Blood and Lampblack • Sorting Notes with MailMerge™ • WordStar® for the Academic • Filling in Forms • Test Your Reaction Time • Too Hot to Handle • Tax Deductions • Double Trouble or Value? •

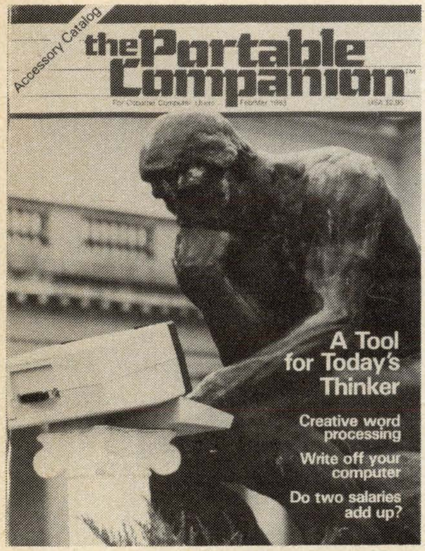
April/May 1983:
Portable, Transportable or Ergonomic Backache? • Syndicating with your Osborne • Duck • Practical Editing: Pt 1 • Graphics Screenwriting • Confessions of an Armchair General • Technical Tips • I/O, I/O, It's Off to Work We Go pt. 3 • Saving Your ASCII •

June 1983:
The Next Logical Step • Self-Publishing • Technical Tips • Rx: Osborne • Graphic Illustrations • Dating Your Computer • MBASIC Brain Twisters • Programming the Serial Port • Practical Editing •

April/May 1984:
Printing the Special Function Keys • Software for Attorneys • Easy Animated Graphics • Portable Psychology • Beyond the Joystick • WordStar, dBASE II and SuperCalc™ Data Sharing • An Osborne On Mount Everest •

WordStar is a registered trademark of MicroPro International. MailMerge is a trademark of MicroPro International. dBASE II is a trademark of Ashton-Tate, Inc. SuperCalc is a trademark of Sorcim. MBASIC is a trademark of Microsoft. CBASIC is a trademark of Digital Research, Inc.

Portable Companion is back!



Subscribe now to the magazine that helps you be more productive with your Osborne computer. $25 for six issues.

☐ Send me **six issues** for $25, payment enclosed, or
☐ Bill me $28 for **six issues**.

☐ **Foreign** (pre-paid): $50 for six issues (airmail), Canada/Mexico $35.

Payment by charge card: ☐ Visa ☐ Mastercard Exp. date ________

Signature required: ____________________ MC ________

Card # ____________________

TOTAL ORDER: $____________

Name ____________________

Company ____________ Title ____________

Address ____________________

City, State, Zip ____________________

☐ Check if you do not want promotional mailings.

Make checks payable to: **Portable Companion**. Offer expires 12/31/84.
Subscription Dept.: *42680 Christy St., Fremont CA 94538.* UG 10
Please allow 6-8 weeks for processing.

Circle number 15 on the reader service card.

New Products

CP/M and MS-DOS computers, software, communication products, networks and accessories.

Free information is available on the products described here. We provide a *reader service card* in this issue with numbers corresponding to each product announcement. Fill out the reader service card and drop it in the mail (please include your address). We forward your address to the product manufacturer or vendor, who sends you the information.

Thanks for filling out the questionaire on the card. This feedback helps us plot an editorial course, and the answers provide demographic information to media planners. The more responses this magazine can generate, the more it can grow in pages and in product coverage. We appreciate your support.

Computers & Systems

16-bit and Lotus For Kaypro Computers

The CO-POWER-88 board by SWP Microcomputer Products, available for Kaypro 10 and 4/84 models, can now run Lotus 1-2-3® (Lotus Development Corp.), IBM PC release 1A. This new utility is now available from SWP for $29.95 retail.

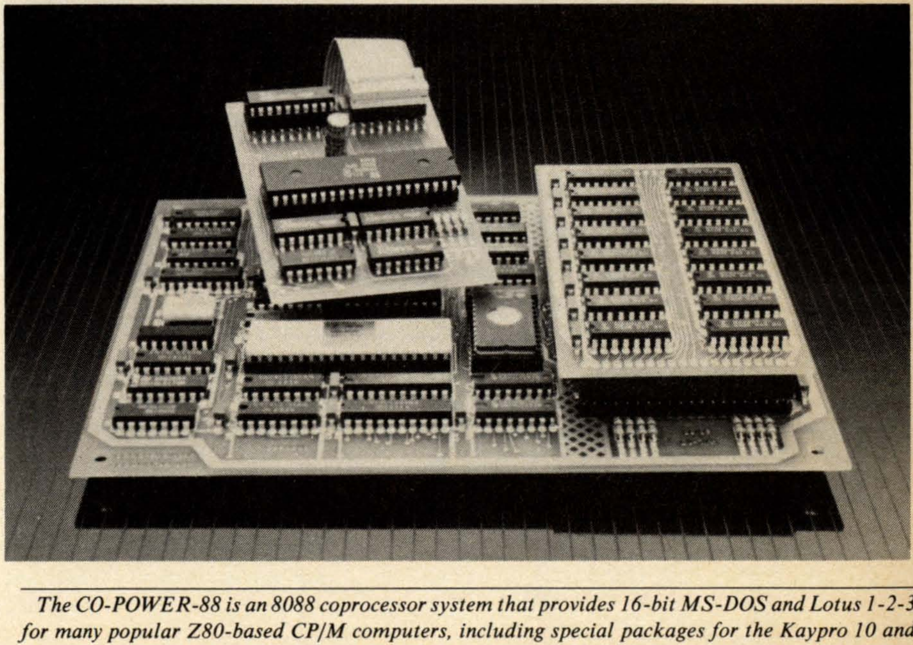
Lotus 1-2-3 is installed as it is on the IBM PC with a monochrome board, with the same limitations as on the IBM PC (e.g., you can't do graphs on the screen, but you can print graphics). The new utility requires SWP's new release of MS-DOS 2.11A. Owners of earlier versions of the CO-POWER-88 system can contact SWP for a disk update.

CO-POWER-88 is an 8088 coprocessor system that can be added to many popular Z80-based CP/M computers. Current versions of the board include a 5.33 MHz 8088 processor and 256K RAM. The newest version of the board, available this fall, that has one megabyte of RAM.

The host computer retains all of its previous features, and while the computer is running CP/M, CO-POWER-88's RAM can be used as a high-speed RAM disk.

SWP Microcomputer Products, 2500 E. Randol Mill Rd. #125, Arlington TX 76011 (817-469-1181).

Circle 51 on your reader service card.



The CO-POWER-88 is an 8088 coprocessor system that provides 16-bit MS-DOS and Lotus 1-2-3 for many popular Z80-based CP/M computers, including special packages for the Kaypro 10 and 4/84 models.

New Apricots

Applied Computer Techniques introduced five new products that extend the capabilities of the Apricot line of computers. The Apricot F1 ($1595) is an 8086-based MS-DOS computer with an infrared keyboard and optional mouse/trackerball, color graphics, 256K RAM expandable to 768K, and a 720K double-sided 3.5" disk drive. The computer uses GSX graphics.

The Apricot F1e ($1295) is an F1 with a 315K disk drive and 128K of RAM. Bundled with the F1e is Dr. Logo from DRI, Personal BASIC, and CP/M-86.

The Apricot Portable ($3195, or $2695 without color) is an 8086-based machine that incorporates voice recognition, a 25-line by 80-column LCD screen, 256K RAM expandable to one megabyte, and a 720K double-sided 3.5" disk drive. The voice recognition software has a vocabulary of up to 4096 words.

The Apricot Point 7 links up to seven Apricots of any model and IBM PCs in a cluster-controlled multi-user system. The Apricot Point 32 is an Omninet-based local area network that can accommodate up to 32 ACT computers in a system with 200 megabytes of mass storage and tape backup.

ACT Computers (North America) Inc., 3375 Scott Blvd. Suite 342, Santa Clara, CA 95051 (408-727-8090).

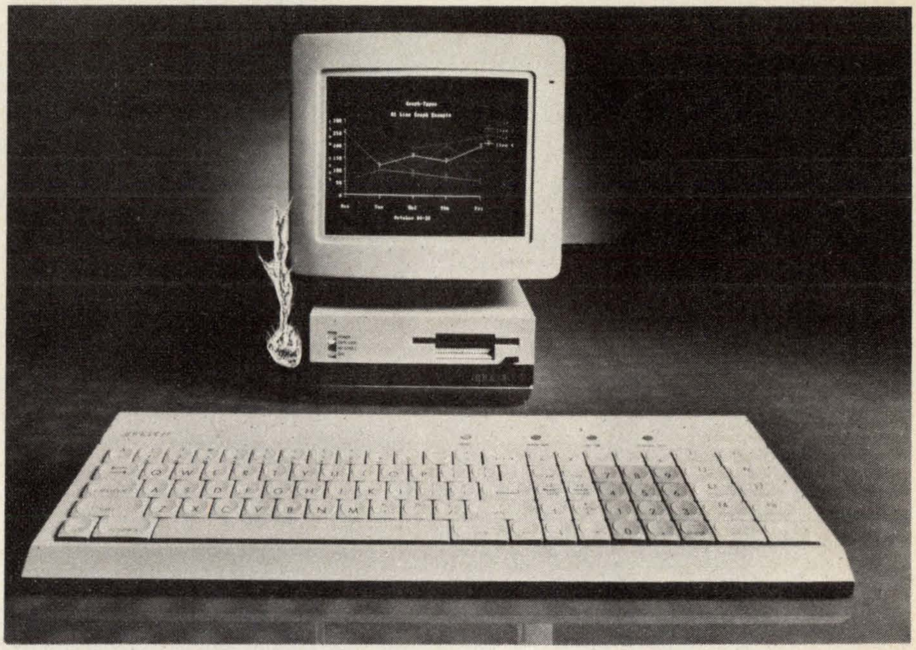
Circle 52 on your reader service card.

PC Compatible Portables

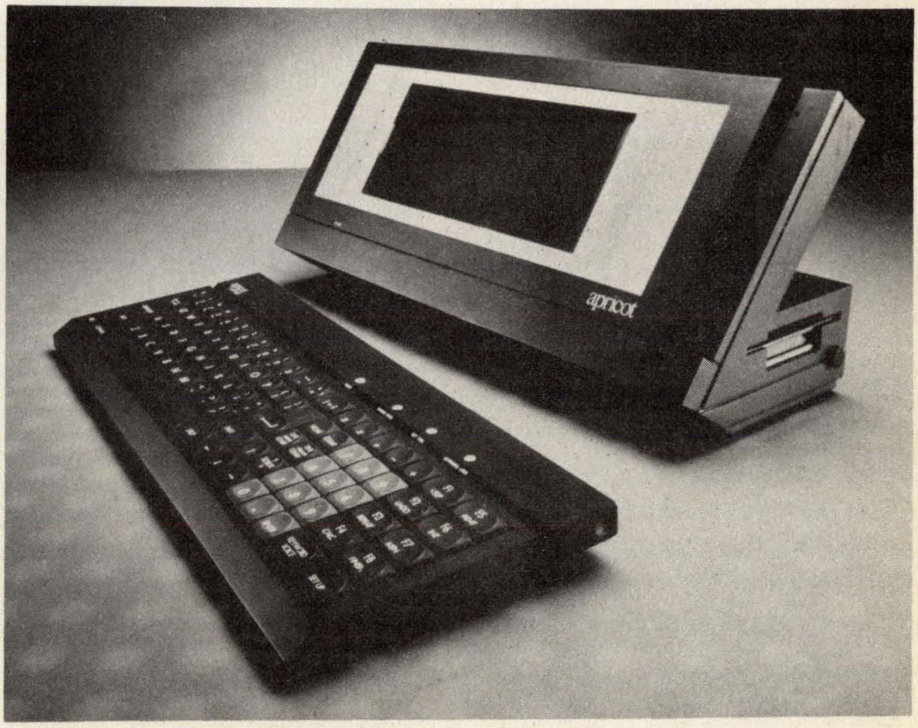
International Quartz Limited of Kowloon, Hong Kong announces three new IBM-PC compatibles featuring Intel's 80186 microprocessor with 128K RAM standard (expandable to 640K), PC compatible 5 1/4-inch 360K double-sided, double-density floppy drives, two RS-232 serial ports, and a Centronics-type parallel interface.

Model 9240 features a 25-line LCD (liquid crystal display), one floppy disk drive, and weighs less than 15 pounds. It operates on AC power only.

Model 9230 features the same 25-line LCD, has two floppy disk drives, a built in 80-column dot matrix printer, and detachable keyboard unit with a three-line LCD that serves as a stand-alone portable word processor for limited computing while traveling. An optional built-in 600 baud modem is used to communicate with the main unit and download data remotely.



The Apricot F1 features an infrared keyboard, color graphics, and an optional mouse/trackerball. The 8086-based computer has 256K RAM (expandable to 768K) and a 720K double-sided 3.5" disk drive. The F1 graphics software can produce images in 640 x 256 pixels with four colors. The F1 can be used with any monitor and is shown here with ACT's 10" color monitor.



The Apricot Portable features voice-recognition, infrared keyboard and mouse/trackerball, and a full-size, 25-line x 80-column LCD screen in a compact, 13-pound package.

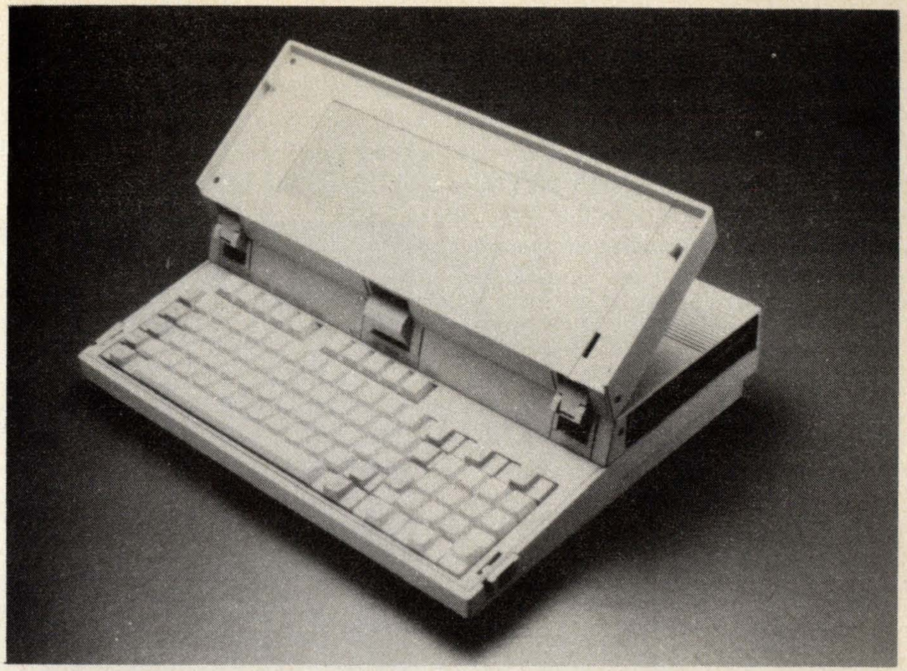
Weight of the full system is under 35 pounds.

The keyboard unit is full size and powered by six AA 1.5 volt rechargeable nickel cadmium batteries. It uses a Hitachi CMOS HD 6303R CPU with 8K CMOS static RAM. An optional RAM/ROM cartridge provides extra memory and storage. The Interquartz Memowriter firmware provides text editing functions and communications with the main unit through the 600 baud serial modem, a real time clock with calendar and user-defined function keys.

Model 9231 is a conventional unit that works with or without the keyboard unit. Using a CRT display and two 360K floppy drives, it weighs under 35 pounds.

In the U.S. contact Jack Carroll, 23505 Crenshaw Blvd., Torrance, CA 90505 (213-539-8944).

Circle 53 on your reader service card.



The Model 9240 (International Quartz Limited) is a PC-compatible portable computer with a 25-line LCD display, 128K RAM expandable to 640K, the 80186 processor, a single 360K double-sided double-density 5.25" disk drive, two RS-232 ports and a Centronics parallel port.

$2000 Hard Disk Computer

Morrow has broken the $2000 price barrier with the introduction of the Morrow MD5-E, a complete computer system with a five-megabyte hard disk that retails for $1999.

The MD5-E uses a 5.4M 5.25" Seagate Technology Winchester drive and a 384K floppy disk drive. Bundled with the package is the NewWord word processing program and Correct-It spelling checker. The system runs CP/M Plus.

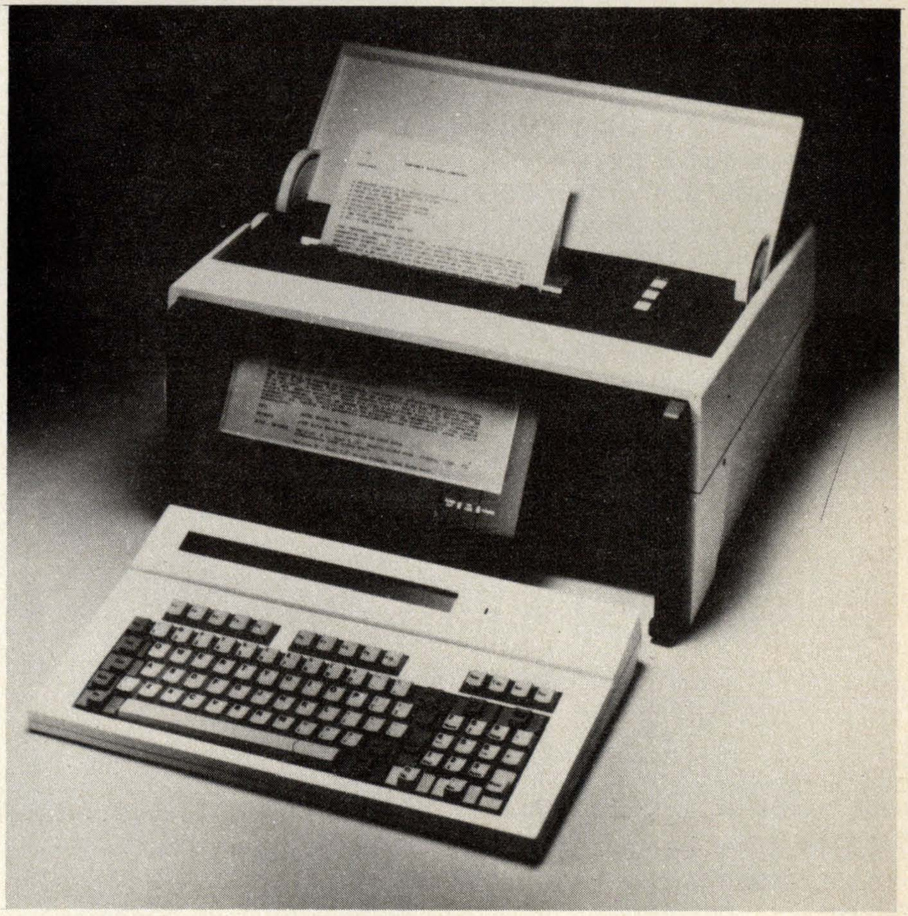
Morrow Inc., 600 McCormick St., San Leandro, CA 94577 (415-430-1970).

Circle 54 on your reader service card.

Software

CP/M Emulator For PC/MS-DOS Computers

Vertex systems announces two major enhancements to 80Mate, the soft-



The Model 9230 (International Quartz Limited) is a PC-compatible transportable computer with a 25-line LCD display, the 80186 processor, a built-in dot matrix printer, a battery-powered detachable keyboard that can operate in stand-alone mode, and two double-density double-sided 5.25" disk drives.

ware emulator that lets an IBM PC/XT or compatible computer to run almost any CP/M-80 program without coprocessor boards or other additional hardware. Version 2.0 includes (1) Z-80 processor emulation as well as 8080 emulation, and (2) video and keyboard emulation for the Apple version of CP/M.

The 80Mate package includes 80Term, a CP/M terminal emulator which converts the PC's video and keyboard into any of several popular CP/M computers or terminals. Version 2.0 includes Apple CP/M terminal emulation. Another new feature allows you to input specific terminal characteristics for less common terminals or computers.

80Mate accurately simulates a CP/M-80 environment using ordinary MS-DOS disks. All CP/M-80 commands are available. Vertex Systems also offers products that help transfer programs and data from CP/M disks to MS-DOS disks. The Xeno-Copy program copies files to and from seventy different disk formats, and the APPLE-Turnover board lets and IBM PC read and write Apple CP/M and AppleDOS disks.

80Mate with 80Term retails for $149.50. Vertex Systems, 6022 W. Pico Blvd., Suite 3, Los Angeles, CA 90048 (213-938-0857).

Circle 55 on your reader service card.

New Wave Outline Processor for CP/M 2.2

Compusophic Systems of Aloha, Oregon claim KAMAS® (Knowledge And Mind Amplification System) is the first and only outline processor for CP/M. This $175 integrated program includes word processing, telecommunications, a programming language, information retrieval and outline processing.

The word processor is a full screen editor with text justification, margin setting, line spacing, pagination and printing output controls.

Telecommunications requires Hayes Smartmodem 1200 or compatible that implements the RS-232 carrier detect line. Use your system as a terminal to dial remote CP/M systems to retrieve text, or set up your own structured electronic bulletin board system.

The threaded code, extensible programming language is FORTHlike with built in data types and control structures. The command-driven language interface lets you develop customized outline processing applications. Programming is not required to use the outline processor.

The outline processor can be used by writers, managers, and other professionals to organize ideas and categorize information into an electronic outline. Levels of the outline can be collapsed off the screen (useful for hiding detail) and then expanded back onto the screen later. You can access and alter text and structures by keyword.

Product support plans include a monthly newsletter (free first year) with application developments for calendars, setting up a remote bulletin board system, a novel approach to cataloguing a record collection based on moods, experiments with expert systems and artificial intelligence, setting up systems for education and training, designing and writing an interactive novel, and more.

Formats available now are Kaypro 2, 2X, 4, and 10, Morrow MD2 and MD3 with MDT60 or Televideo 900 series terminal, Epson QX-10. Minimum requirements are CP/M 2.2 on Z80 computer with 64K RAM and two 183K floppy disk drives. An 80 column screen and a printer are recommended. From Compusophic Systems, PO Box 5549, Aloha, OR 97007 (503-649-3765).

Circle 56 on your reader service card.

Fixed Assets Accounting

Depreciation-Master II from Generic Software (Marquette, MI) provides depreciation figures for capital assets using up to three depreciation methods per asset. A report in tax form 4562 format is provided for federal tax rule requirements, and a selective depreciation summary report is provided for internal asset accounting. Federal tax rules, internal tax methods, and separate state methods can be accounted for in the same asset database file.

Depreciation-Master II supports all standard depreciation methods and schedules and is suitable for CPA firms, banks and businesses with CP/M-80 and 64K RAM, MS-DOS and 128K RAM, and IBM-PC and 96K RAM computers. MBASIC, two disk drives, an 80-column display and line printer are required.

Available disk formats include Zenith Z100, Heath/Zenith Z90, Kaypro II/4, Epson QX-10, and most other CP/M-80 systems for $149, or IBM-PC (and compatibles) for $199. From Generic Software, PO Box 790, Marquette, MI 49855 (906-249-9801).

Circle 57 on your reader service card.

16-bit Magic

MagicPrint, MagicBind and MagicIndex are now available for CP/M-86 and MS-DOS (including IBM PC-DOS 1.1 and 2.0) 16 bit computers.

MagicPrint ($195) works with other word processors (WordStar, Electric Pencil, Magic Wand, and Select) to give you special formatting capabilities such as true proportional space printing, up to 15 automatic footnotes per page, multi-column printing and other features.

MagicBind ($250) is a file merging program which doesn't choke on empty fields, automatically numbers chapters, sections or paragraphs, and incorporates all the MagicPrint functions.

MagicIndex (includes all features of MagicPrint and MagicBind for $295) compiles an index and table of contents.

All programs are written in assembly language. The programs require 48K RAM for the 8-bit CP/M versions, and standard daisywheel printer or NEC Spinwriter. Requires 90K RAM for 16-bit version.

Circle 58 on your reader service card.

Real Estate Planning

TaxCalc Software offers a real estate investment planning template for more than a dozen popular spreadsheet programs including Lotus 1-2-3, SuperCalc, Multiplan and VisiCalc.

The template performs calculations that help evaluate the true worth of an investment over one to ten year periods. It also performs detailed cash flow analysis, taking taxes and other variables into account. The planner considers these and other factors in determining the value of an investment: inflation, investor's tax bracket, vacancy rates, rental expenses, rent increases, taxes and fair market value.

The TaxCalc Real Estate Planner is $100. TaxCalc Software, Inc., 4210 West Vickery Blvd., Fort Worth, TX 76107 (817-738-3122).

Circle 59 on your reader service card.

Marina Management

Office Computer Systems announces a computer billing program for marinas. The program calculates electric rates and does complete billing and accounts receivable for marinas and yacht clubs.

The Marina Program maintains a list of names, addresses and telephone numbers of berth rental tenants. The program stores information about the rental, the current and previous electric meter readings, current billing amounts and balance due. The $500 program requires CBASIC-80 and CP/M-80, or CBASIC-86 and CP/M-86 (demo disk is $25 applied to purchase price).

Office Computer Systems (division of Computer Medalytics), P.O. Box 91, Corte Madera, CA 94925 (415-453-9960).

Circle 60 on your reader service card.

Checks

CHECKS is a $69.95 check writing program for small businesses and individuals. The program does batch printing, sorts check entries according to date, class or account name, compares the check register to your bank statements, and maintains creditors' names, addresses and account information.

The program requires a Z80-based CP/M computer with 64K RAM, or an MS-DOS computer with 128K RAM.

Future Communications, 210 West Benson St., Decatur, GA 30030 (404-373-4831).

Circle 61 on your reader service card.

Analytical Research

DBi Software Products announced TRAJECTORIES, a comprehensive statistical processing software system for CP/M, MS-DOS or PC-DOS computers with at least 64K RAM. TRAJECTORIES includes 40 programs developed by statisticians to perform regression analysis. The package includes an interface to mainframe computers.

DBi Software Products, One Energy Place, 5805 East Pickard Road, Mt. Pleasant, MI 48858 (517-772-5055).

Circle 62 on your reader service card.

Hard Disk Backup

Stok Software offers its BackRest program with DRI's Concurrent DOS for performing hard disk backup. The new version of BackRest backs up and restores both PC-DOS and CP/M-86 files in one single process (Concurrent DOS can handle both file structures at the same time). Other versions of BackRest run under CP/M 2.2, CP/M Plus, MP/M, CP/M-86, TurboDOS, MS-DOS, PC-DOS and local area networks such as PC/NET and OMNI/NET.

Stok Software, 17 West 17th Street, New York, NY 10011 (212-243-1444).

Circle 63 on your reader service card.

File Cataloguing

Computer Thaumaturgy announces the LOG System for computers using PC-DOS, MS-DOS, CP/M-80, or CP/M-86. LOG helps you keep track of disk files. LOG stores a 95-character description about each file, along with creation and revision dates and times. You can also create a catalog of files already in use.

With LOG you can merge and print sorted lists of files from multiple disks. On-line help is available to make the program easy to learn. The LOG System is $44.95.

Computer Thaumaturgy, 1212 Miami Valley Tower, 40 West Fourth St., Dayton, OH 45402 (513-461-2126).

Circle 64 on your reader service card.

Application Development

The Software Store announces INFO-80, an application development system with data base management capabilities that is designed for beginners as well as advanced users.

INFO-80 has building blocks for file definition, data entry and editing, file sorting and merging, record access and selection, report and letter writing, file format conversion and menu-driven job control. INFO-80 files can be merged with text for form-letter processing. INFO-80 costs $395 and is available for most 8080, Z80 and 8085 computers with 64K RAM and CP/M.

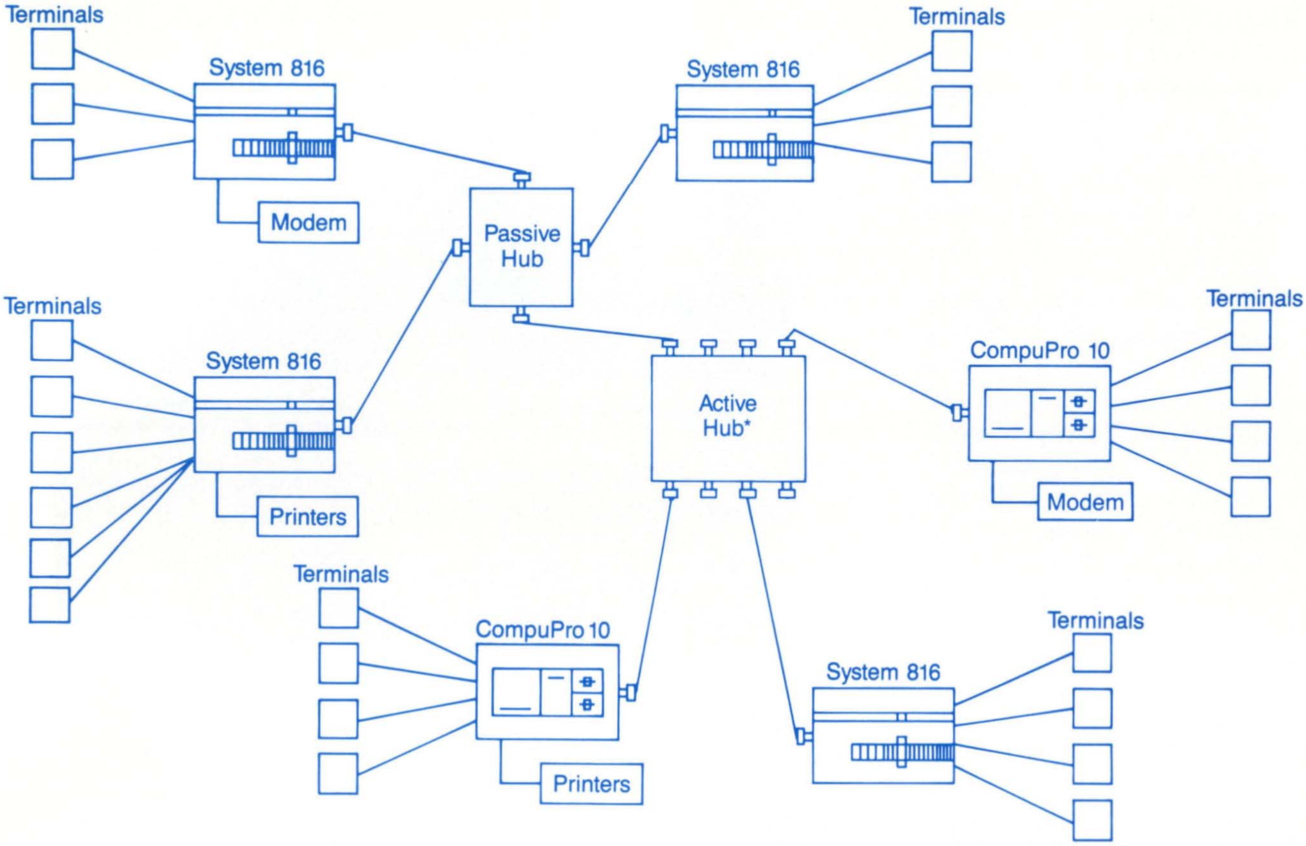
The Software Store, 706 Chippewa Square, Marquette, MI 49855 (906-228-7622).

Circle 65 on your reader service card.

Write Micro Software on VAX

Modula-2/VX86 from Logitech Inc. (Redwood City, CA) is the first cross development system to be used to produce software for the 8086/8088 microcomputers on Digital Equipment Corporation's VAX/VMS system.

Modula-2/VX86 is a complete implementation of the Modula-2 language as defined by Niklaus Wirth.



*Active hub required for more than four nodes. Maximum number of nodes is 255 in any combination of System 816s and CompuPro 10s.

The S-100 Bus Network from CompuPro uses Datapoint's ARCnet protocol and requires a passive hub to connect up to four CompuPro System 816 computers, or an active hub to connect up to 255 CompuPro System 816 computers or other computers supporting the ARCnet protocol.

Used to develop software for 8086-based computers that run MS-DOS or CP/M-86, it comes with a full standard library of modules and generates high-speed native code.

It supports the 8087 numeric coprocessor and includes a complete set of development tools including data communication and conversion utilities for uploading and downloading programs. Priced at $4,500 each, the VX86 system can also generate code to be stored in read-only memory (ROM code). Logitech Inc., 805 Veterans Blvd., Redwood City, CA 94063. (415) 365-9852

Network Systems

Network for CompuPro

CompuPro introduced an IEEE 696 (S-100) compatible network board for its System 816 multi-user computers. The network board supports Datapoint's ARCnet protocol, and requires only a "passive hub" and RG62 coaxial cable to connect up to four System 816 multi-user systems ("nodes").

With the addition of an "active hub," the board can connect up to 255 System 816 "nodes" over a distance of four miles. Computers that support ARCnet (including CompuPro 10, IBM, Compaq, Texas Instruments, S-100 bus computers and others) can be integrated into one network. The board runs under CompuPro's version of Concurrent DOS 8-16, the multi-user operating system for sixteen-bit systems. The network uses a 2.5-megabit-per-second data transfer rate.

The S-100 bus network board retails for $395. CompuPro, 3506 Breakwater Court, Hayward, CA 94545 (415-786-0909).

Circle 66 on your reader service card.

Communications

Smartcom Now Has XMODEM

Hayes Microcomputer Products has released Smartcom II 2.0, an upgrade of its telecommunications software that now includes the popular XMODEM communications protocol developed in the public domain. The update also includes VT52 and T102/100 terminal emulation and batch commands for automatic transmissions.

The Smartcom II package ($149) works well with the Hayes Smartmodem. Smartcom II 2.0 protects passwords, tests the modem without going on-line, and lets you switch from voice to data transmission during the same phone call.

The XMODEM protocol lets you set up error-free communications with many bulletin board systems and public domain software exchanges. XMODEM is used in many other communications programs.

Hayes Microcomputer Products, 5923 Peachtree Industrial Blvd., Norcross, GA 30092 (404-449-8791).

Circle 67 on your reader service card.

Counseling by Telecom

TelePsych is a new computer telecommunication service that provides consultation with a clinical psychologist. Timothy Miller, Ph.D., a licensed clinical psychologist, makes consultation available 24 hours a day to anyone with a personal computer and a modem.

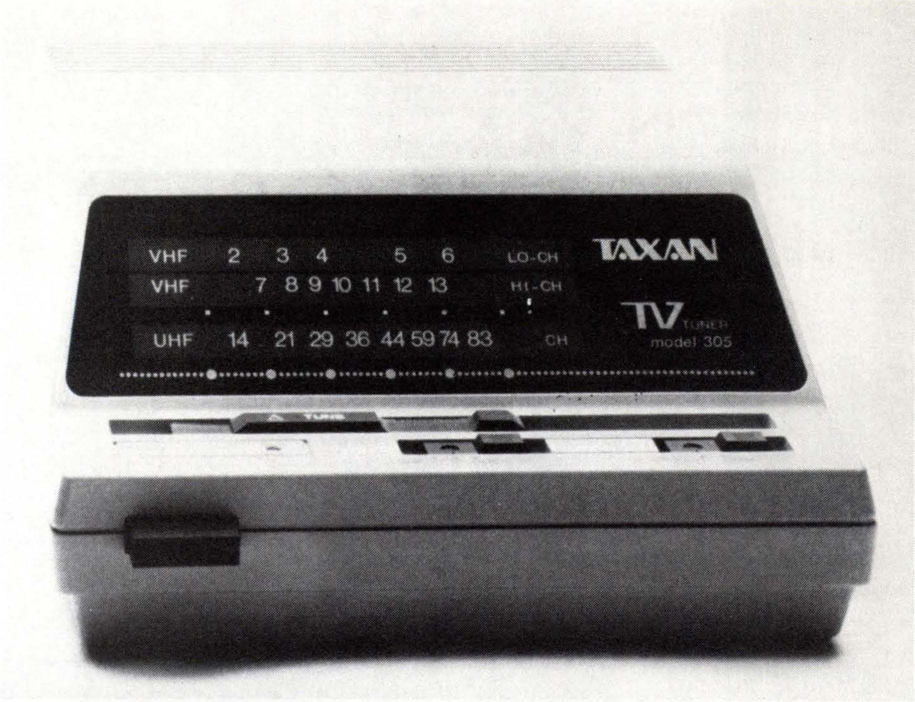
An enrolled client prepares his or her message on a personal computer, then calls TelePsych to transmit the message. Dr. Miller's response is prepared within seven days, and the client calls back to receive the reply. All responses are prepared by Dr. Miller, who is a human being. All messages are treated as confidential. Clients are billed 7/10 of a cent per word sent to Dr. Miller and 2 cents per word in Dr. Miller's replies.

Use your modem and computer to call: 209-473-8296. No voice line available. For appointment call by modem, or write to Timothy Miller, Ph.D., 2155 W. March Ln., 2B, Stockton, CA 95207.

Circle 68 on your reader service card.

Semiconductor News

NewsNet now makes available the Semiconductor Industry & Business Survey (SIBS), which provides business and marketing information on the semiconductor industry. SIBS is one of more than two dozen publications on NewsNet covering electronics and computers. Regular departments in SIBS are company profiles, reviews, business and marketing trends, delivery and price trends, technology and licensing agreements, organizational developments, new expansions and industry rumors. SIBS is published by HTE Management.



TAXAN model 305 television tuner.

NewsNet began commercial operation in April 1982. For more information: NewsNet, 945 Haverford Rd., Bryn Mawr, PA 19010 (215-527-8030 or 800-345-1301).

Circle 69 on your reader service card.

Accessories

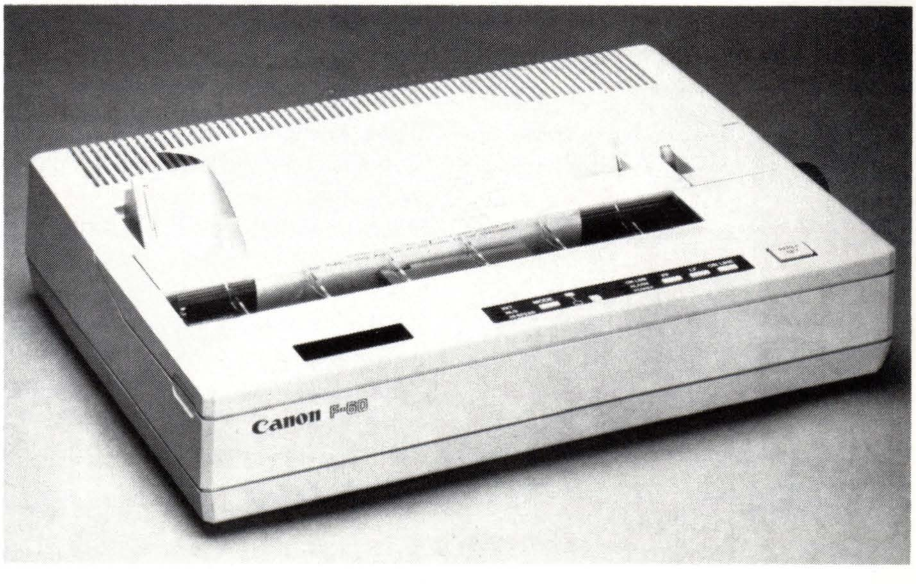
Thermal Transfer Printer

Canon expands its printer line to include a quiet thermal transfer printer, the F-60, which produces letter-quality printing on regular paper. The second printer is an inexpensive, quiet thermal printer (the T-20), which uses special thermal paper.

The F-60 produces sharp, clear characters and is capable of producing high-resolution graphics up to 360 dots per inch. The T-20 prints a 5x7 dot matrix character.

Canon USA, Printer Division, One Canon Plaza, Lake Success, NY 11042.

Circle 70 on your reader service card.



Color Monitor With Audio Becomes... High Resolution Color TV

Taxan Corporation (City of Industry, CA) began shipments of their model 305 television tuner in June at $129.95 retail.

The tuner allows any brand composite monitor with audio to become a color TV with a clearer picture than regular televisions. Sold in computer stores or contact TAXAN Corp., 18005 Cortney Court, City of Industry, CA 91748 (818-810-1291).

Circle 71 on your reader service card.

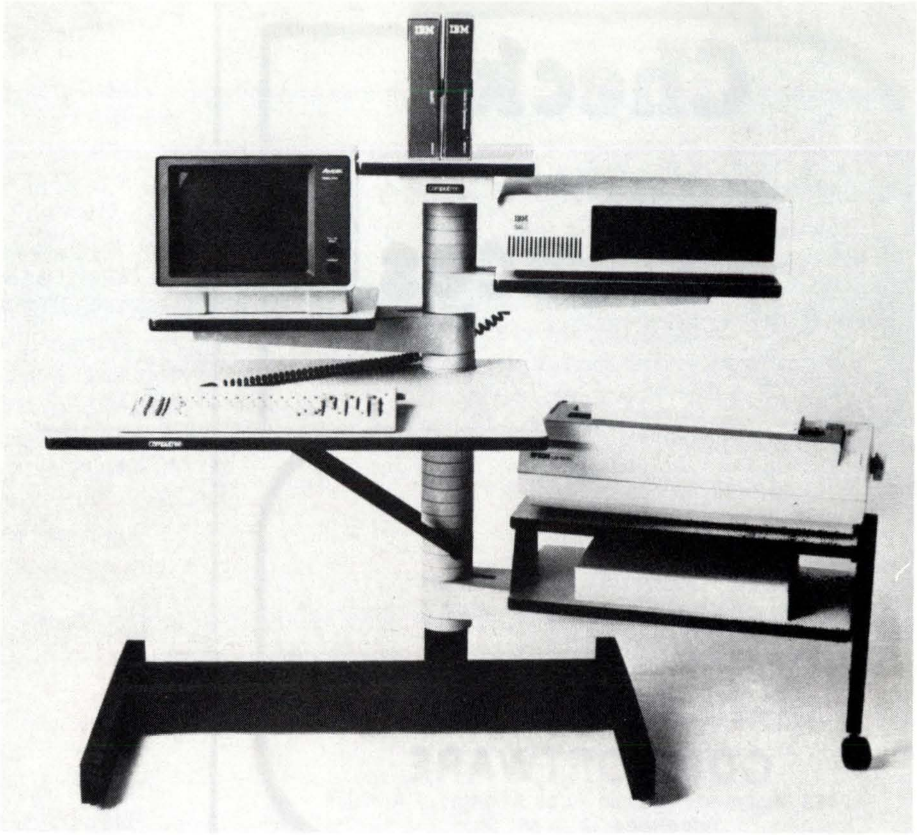
Computer Furniture

Frontline Products announces COMPUTREE, a support stand for electronic equipment that uses a center pole acting as a pivot for tables, arms, spacers and a base. COMPUTREE can easily accommodate an entire computer system including display, keyboard, printer, paper and media.

The tables attached to a center pole can move up and down, swivel in and out and rotate in a 360-degree angle. The tables are high-pressure laminate pionite, and the 3"-diameter center pole is steel tubing.

Frontline Products, 1916 W. Glenoaks Blvd., Glendale, CA 91201 (818-843-1860).

Circle 72 on your reader service card.



The COMPUTREE is a modular unit with a center pole used as a pivot for tables and arms to hold computer equipment.

Power Problems?

A four-page brochure that explains typical power problems with computers is available from Shape Magnetronics, Inc., manufacturer of Line Tamer power conditioners. The brochure also shows price/performance comparisons of available solutions.

For a copy write to Shape Magnetronics, 901 DuPage Ave., Lombard, IL 60148 (312-620-8394).

Circle 73 on your reader service card.

Events

Third Annual Pacific Northwest Computer Graphics Conference

October 29 & 30 1984 the University of Oregon in Eugene hosts this two day event at the Hult Center for the Performing Arts and Conference Center/Hilton complex. A non-commercial exhibit of computer graphics works, a monday evening film and video show and six general session presentations accompany an extensive trade exhibition. Session leaders include Professor Donald Greenberg on architectural simulation, Professor Tom Linehan on computer animation, John Briggs on real-time visual simulation applications, Steve Putz on the use of graphics in the Smalltalk-80 language system, Ted Driscoll on earth resources imaging and analysis applications, and Gene Youngblood, who produced and will narrate Monday's film and video show. Computer Graphics Program, University of Oregon, Continuation Center, 333 Oregon Hall, Eugene, OR 97403 (503-686-4231).

Forth Convention

The 6th Annual Forth Convention will be held Nov. 16-17 at the Hyatt Palo Alto in Palo Alto, California. Sponsored by the Forth Interest Group (FIG), the convention will focus on the latest developments and applications of the Forth computer language. The FIG is a non-profit member-supported organization with over 4800 members worldwide. FIG membership is $15/year and includes a subscription to the bi-monthly *FORTH Dimensions*. For additional information call 415-962-8653 or write FIG, P.O. Box 1105, San Carlos, CA 94070.

Checks & Balances

Setting the standard in
checkbook and budget management
for home and office.

NEW! Rolodex function for names,
addresses, and phone numbers!

•Display, add and change entries with full on-screen editing options•
•Single entry system•
•Unlimited categories•
•Divide entries between multiple categories•
•Flexible search and printing•
•Commands in plain English•
•
For CP/M-80 2.2, MS-DOS and PC-DOS
•
Only $74.95, with CHECK PRINTING & ROLODEX
Visa and MasterCard accepted
Dealer/Distributor/Bundling rates available

CDE SOFTWARE
2463 McCready Avenue • Los Angeles, CA 90039
Telephone: (213) 661-2031

Circle number 3 on the reader service card.

Editorial *continued from page 4.*

The Turbo Pascal tutorial starts at the "ground" and presents a simple Pascal program. Pascal is, in our opinion, easier to learn than most other computer languages.

"Running MBASIC Programs" assumes no experience with MBASIC and teaches anyone how to use the MBASIC interpreter to run programs already written by others.

The "CBASIC Compiler" article gives a professional opinion on the merits of using the CBASIC compiler for real business applications. CBASIC is one of the oldest personal computer languages and the first to be used for commercial CP/M business applications.

"Batch Processing" shows you how to make CP/M or MS-DOS perform sets of commands automatically, and is perhaps the first taste of programming for most users.

Amateur programmers, don't hide behind excuses for not showing up at parties, and don't keep your dBASE II or Pascal or BASIC adventures to yourself. There is no prerequisite for programming. You don't need a "mathematical" or even a "logical" mind. You don't need to be the typical white male nerd. We know an excellent programmer who flunked Logic and Calculus, dropped out of school, moved to Silicon Valley and eventually made a fortune in the personal computer industry.

But now she's no amateur.

Advertiser Index

Service	Page
1. Awareco	33
2. Baen Books	65
3. CDE Software	76
4. Celestial Arts	66
5. CompuPro	C4
6. Compusophic Systems	61
7. Echelon	5
8. EWDP Software	51
9. FOG (First Osborne Group)	63
10. Mendocino Software	17
11. Micah	15
12. Paradigm Consultants	49
13. Orange Microware	51
14. Poor Person Software	15
15. Portable Companion	67
16. SBT	21

It's *COMDEX Time!*

Advertise in the next **double** issue of ***User's Guide*** magazine (issue 11/12), and get **EXTRA Distribution** at the **Fall COMDEX Show**. Reserve your space today! The deadline is October 3rd. Materials are due October 10th. Call 415-364-0108.

Advertiser Services

Send all camera-ready materials to:
User's Guide Production
2055 Woodside Road, Suite 180
Redwood City, CA 94061

Jay Eisenberg - Advertising Director
JE Publishers' Representative Company
6855 Santa Monica Blvd.
Los Angeles, CA 90038
(213) 467-2266
Boston - (617) 437-7628 Chicago - (312) 445-2489
Dallas - (214) 660-2253 Denver - (303) 595-4331
New York - (212) 724-7767 San Francisco - (415) 864-3252

Please enroll me at the low introductory offer
CHECK ONE:
☐ Send me six issues for $18, payment is enclosed, or
☐ Bill me for six issues at $21, or
☐ Send me twelve issues for $34, payment is enclosed, or
☐ Bill me for twelve issues at $40, or
☐ Foreign rates, payment enclosed: Canada & Mexico $22/six issues, overseas $40/six issues (airmail).

Payment by charge card: ☐ Visa ☐ Mastercard

Signature required: ______________________ Exp. date _____ MC ___

Card # ______________________________

TOTAL ORDER: $ _______

Name ______________________________

Company ______________ Title ______________

Address ______________________________

City, State, Zip ______________________________
☐ Check if you do not want promotional mailings.

Make checks payable to: User's Guide, Offer expires 12/31/85
Subscription Dept.: P.O. Box 5245, Redwood City, CA 94063.
Please allow 6-8 weeks for processing.

UG 10

Please enroll me at the low introductory offer
CHECK ONE:
☐ Send me six issues for $18, payment is enclosed, or
☐ Bill me for six issues at $21, or
☐ Send me twelve issues for $34, payment is enclosed, or
☐ Bill me for twelve issues at $40, or
☐ Foreign rates, payment enclosed: Canada & Mexico $22/six issues, overseas $40/six issues (airmail).

Payment by charge card: ☐ Visa ☐ Mastercard

Signature required: ______________________ Exp. date _____ MC ___

Card # ______________________________

TOTAL ORDER: $ _______

Name ______________________________

Company ______________ Title ______________

Address ______________________________

City, State, Zip ______________________________
☐ Check if you do not want promotional mailings.

Make checks payable to: User's Guide, Offer expires 12/31/85
Subscription Dept.: P.O. Box 5245, Redwood City, CA 94063.
Please allow 6-8 weeks for processing.

UG 10

LAST CHANCE TO SAVE 25% off newsstand price!

Subscribe Now

before 1985 new rates!

Please enroll me at the low introductory offer
CHECK ONE:
☐ Send me six issues for $18, payment is enclosed, or
☐ Bill me for six issues at $21, or
☐ Send me twelve issues for $34, payment is enclosed, or
☐ Bill me for twelve issues at $40, or
☐ Foreign rates, payment enclosed: Canada & Mexico $22/six issues, overseas $40/six issues (airmail).

Payment by charge card: ☐ Visa ☐ Mastercard

Signature required: ______________________ Exp. date _____ MC ___

Card # ______________________________

TOTAL ORDER: $ _______

Name ______________________________

Company ______________ Title ______________

Address ______________________________

City, State, Zip ______________________________
☐ Check if you do not want promotional mailings.

Make checks payable to: User's Guide, Offer expires 12/31/85
Subscription Dept.: P.O. Box 5245, Redwood City, CA 94063.
Please allow 6-8 weeks for processing.

UG 10

No Postage
Necessary
if mailed
in the U.S.

BUSINESS REPLY CARD
FIRST CLASS PERMIT #36 REDWOOD CITY, CA

POSTAGE WILL BE PAID BY ADDRESSEE

User's Guide
PO Box 5245
Redwood City, CA 94063

Billing Address:

____

____

____

No Postage
Necessary
if mailed
in the U.S.

BUSINESS REPLY CARD
FIRST CLASS PERMIT #36 REDWOOD CITY, CA

POSTAGE WILL BE PAID BY ADDRESSEE

User's Guide
P.O. Box 5245
Redwood City, CA 94063

Please send me the following back issues:

Payment by charge card: ☐ Visa ☐ Mastercard Payment must accompany order.

Signature required: __________ Exp. date ______ MC ______

Card # ____________________

☐CP/M Summary Card, $2 each: # of cards ______ = $______

☐Back issue #1 at $5 each, issue #2 at $6 each, and all other issues at $4 each:

Issue# 1 Qty. ______ at $5 each ($6.50 1stC or foreign, $11 air) = $______

Issue# 2 Qty. ______ at $6 each ($7.50 1stC or foreign, $12 air) = $______

Issue# ___ Qty. ______ at $4 each ($5.50 1stC or foreign, $10 air) = $______

TOTAL ORDER: $______

Name ____________________

Company __________ Title __________

Address ____________________

City, State, Zip ____________________

☐ Check if you do not want promotional mailings.

Make checks payable to: User's Guide, Offer expires 12/31/85
Subscription Dept.: P.O. Box 5245, Redwood City, CA 94063.
Please allow 6-8 weeks for processing.

UG 10

Act now to complete your set of *User's Guide* back issues! Limited offer while supplies last! This may be your last chance to get a complete set!

Billing Address:

____

____

____

No Postage
Necessary
if mailed
in the U.S.

BUSINESS REPLY CARD
FIRST CLASS PERMIT #36 REDWOOD CITY, CA

POSTAGE WILL BE PAID BY ADDRESSEE

User's Guide
P.O. Box 5245
Redwood City, CA 94063

Billing Address:

____

____

____

No Postage
Necessary
if mailed
in the U.S.

BUSINESS REPLY CARD
FIRST CLASS PERMIT #36 REDWOOD CITY, CA

POSTAGE WILL BE PAID BY ADDRESSEE

User's Guide
P.O. Box 5245
Redwood City, CA 94063

No Postage
Necessary
if mailed
in the U.S.

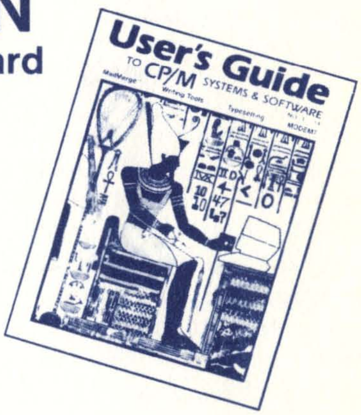
BUSINESS REPLY CARD
FIRST CLASS PERMIT #36 REDWOOD CITY, CA

POSTAGE WILL BE PAID BY ADDRESSEE

User's Guide
PO Box 5245
Redwood City, CA 94063

FREE INFORMATION

User's Guide Reader Service Card



Here is an easy way for you to get manufacturers' information about products advertised or mentioned. Circle the numbers on the card that correspond to key numbers at bottom of advertisements and fill out and mail the postage paid card and the literature will be sent to you free of charge.

No Postage
Necessary
if mailed
in the U.S.

BUSINESS REPLY CARD
FIRST CLASS PERMIT #36 REDWOOD CITY, CA

POSTAGE WILL BE PAID BY ADDRESSEE

User's Guide
PO Box 5245
Redwood City, CA 94063

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

Please complete:

Are you a subscriber? ☐ Yes ☐ No

A. Your computer experience?
1.☐ Novice 3.☐ Intermediate-advanced
2.☐ Semi-intermediate 4.☐ Advanced

C. Use which computer(s)?
a. ☐ Apple
b. ☐ IBM PC
c. ☐ CompuPro
d. ☐ DEC
e. ☐ Osborne
f. ☐ S-100
g. ☐ Radio Shack
h. ☐ Heath/Zenith
i. ☐ TeleVideo
j. ☐ Sanyo
k. ☐ Kaypro
l. ☐ Compaq
m. ☐ Morrow
n. ☐ Columbia
o. ☐ ______________
p. ☐ PC-compat.

F. Where do you use your computer?
1. ☐ Small business
2. ☐ Profession
3. ☐ School/training
4. ☐ Construction
5. ☐ Medium business
6. ☐ Home
7. ☐ Corporation
8. Other: ______________

B. Occupation?
1.☐ Executive/manager
2.☐ Small bus. owner
3.☐ Professional
4. ☐ Other

D. Interested in which software?
a. ☐ Word processing
b. ☐ Accounting
c. ☐ Communications
d. ☐ Data base mgmt.
e. ☐ Medical/Dental
f. ☐ Financial planning
g. ☐ Other vertical
h. ☐ Scientific

E. Interested in which products?
a. ☐ Printers
b. ☐ Modems
c. ☐ Hard disks
d. ☐ Publications
e. ☐ Information services
f. Other: ______________

G. Your immediate need?
1. ☐ Adding peripherals to present system
2. ☐ Upgrading system performance
3. ☐ Using information/communications services
4. ☐ Supplies and accessories

Name ______________________________ **Age** ________ **Sex** _____

Company ______________________________ **Title** ______________________________

Address ______________________________ **City, State, Zip** ______________________________

UG 10

FREE INFORMATION

User's Guide Reader Service Card

Each advertisement and product announcement in this issue has a reader service number. To receive mail from the manufacturer, simply circle the same number on this reader service card, and drop the card in the mail. Your name and address is sent on to the appropriate manufacturer, who sends the information.

The service helps you get detailed information about products you may want to purchase. *User's Guide* does not sell or recommend products associated with this service.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

Please complete:

Are you a subscriber? ☐ Yes ☐ No

A. Your computer experience?
1.☐ Novice 3.☐ Intermediate-advanced
2.☐ Semi-intermediate 4.☐ Advanced

C. Use which computer(s)?
a. ☐ Apple
b. ☐ IBM PC
c. ☐ CompuPro
d. ☐ DEC
e. ☐ Osborne
f. ☐ S-100
g. ☐ Radio Shack
h. ☐ Heath/Zenith
i. ☐ TeleVideo
j. ☐ Sanyo
k. ☐ Kaypro
l. ☐ Compaq
m. ☐ Morrow
n. ☐ Columbia
o. ☐ ______________
p. ☐ PC-compat.

F. Where do you use your computer?
1. ☐ Small business
2. ☐ Profession
3. ☐ School/training
4. ☐ Construction
5. ☐ Medium business
6. ☐ Home
7. ☐ Corporation
8. Other: ______________

B. Occupation?
1.☐ Executive/manager
2.☐ Small bus. owner
3.☐ Professional
4. ☐ Other

D. Interested in which software?
a. ☐ Word processing
b. ☐ Accounting
c. ☐ Communications
d. ☐ Data base mgmt.
e. ☐ Medical/Dental
f. ☐ Financial planning
g. ☐ Other vertical
h. ☐ Scientific

E. Interested in which products?
a. ☐ Printers
b. ☐ Modems
c. ☐ Hard disks
d. ☐ Publications
e. ☐ Information services
f. Other: ______________

G. Your immediate need?
1. ☐ Adding peripherals to present system
2. ☐ Upgrading system performance
3. ☐ Using information/communications services
4. ☐ Supplies and accessories

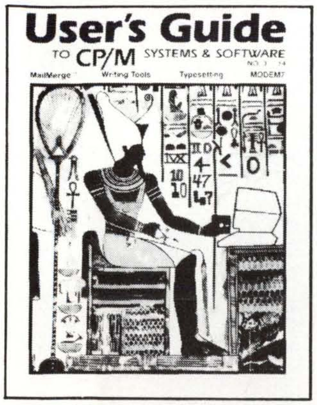
Name ______________________________ **Age** ________ **Sex** _____

Company ______________________________ **Title** ______________________________

Address ______________________________ **City, State, Zip** ______________________________

UG 10

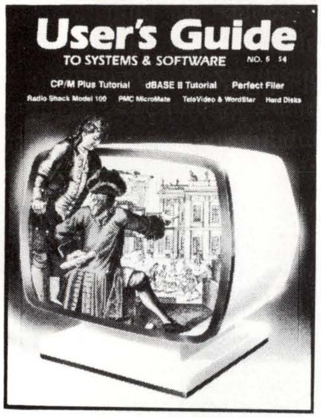
6 TRAINING PACKAGES FOR $18


User's Guide
TO CP/M SYSTEMS & SOFTWARE
MailMerge
Writing Tools
Typesetting
MODEM7

A subscription to ***USER'S GUIDE*** magazine provides the training you need to start using CP/M and MS-DOS systems (including the IBM Personal Computer).

USER'S GUIDE magazine contains self-teaching tutorials and hands-on application notes specific enough to be useful in your applications.

The editors of ***USER'S GUIDE*** evaluate software products while describing how to use them, providing you with the facts you need to make informed buying decisions.


User's Guide
TO SYSTEMS & SOFTWARE
NO. 6 $4
CP/M Plus Tutorial
dBASE II Tutorial
Perfect Filer
Radio Shack Model 100
PMC MicroMate
TeleVideo & WordStar
Hard Disks

> "So far I haven't found one [issue] that wasn't worth more than what it costs. Recommended."
>
> — Jerry Pournelle, ***Byte*** Dec. 1983

User's Guide
Advanced WordStar
COMMX vs. MITE
Perfect Writer vs. WordStar

USER'S GUIDE magazine features well written articles that help you get the most out of your computer investment. For example:

- How to get more mileage out of popular MS-DOS and CP/M programs like WordStar,® SuperCalc™ and dBASE II.™
- How to gain access to thousands of **free** programs in the public domain using a program called MODEM7.
- How to recover data from a damaged disk.
- How to set up and use a small business accounting system.

Make your computer investment profitable for only $18 a year.

☐ Send me six issues for $18, payment enclosed, or ☐ Bill me $21.
☐ Foreign (pre-paid): Canada & Mexico $22/six issues. overseas $40/six issues (airmail).
We accept Visa and Mastercard. Phone: (415) 364-0108

Name ____________________

Company ____________________ Title ____________________

Address ____________________

City, State, Zip ____________________

☐ Check if you do not want promotional mailings.

User's Guide

THE MAGAZINE FOR IBM PC & CP/M SYSTEM USERS

CP/M is a trademark of Digital Research.
WordStar is a trademark of MicroPro
SuperCalc is a trademark of SORCIM Corp.
dBASE II is a trademark of Ashton-Tate.
User's Guide is a trademark of TUG, Inc.

Make checks payable to: User's Guide. Offer expires 11/30/84.
Subscription Dept.: P.O. Box 5245, Redwood City, CA 94063.
Please allow 6-8 weeks for processing.

UG 10

HERE TODAY HERE TOMORROW

When buying a computer, you can't limit yourself to just satisfying today's needs. **The best value in a system comes from its productivity . . . both for today and tomorrow.** CompuPro's System 816™ computer has that value. With all the power and capacity to handle your needs now and down the road.

System 816's longevity stems from top quality components . . . high storage capacity . . . the flexibility to handle a large variety of applications . . . and the speed to get the job done fast. Upgrading is easy, and when it's time to expand from single to multi-user operation, it's as simple as plugging in boards and adding terminals. Your system grows as you grow.

CompuPro also provides a library of the most popular software programs with your system and because it's CP/M® based, you have more than 3,000 other programs to choose from.

Even our warranty is for today and tomorrow. It spans 365 days — and includes the additional security of Xerox Americare™ on-site service nationwide for designated systems.

What's more, CompuPro is one company you can count on to be around tomorrow. For more than ten years we've been setting industry standards, increasing productivity and solving problems.

For a free copy of our business computer buyer's primer, and the location of the Full Service CompuPro System Center nearest you, call **(415) 786-0909 ext. 206.**

CompuPro's System 816. The computer that's just as essential tomorrow as it is today.

CompuPro®
A *GODBOUT* COMPANY
3506 Breakwater Court, Hayward, CA 94545

System 816 and The Essential Computer are trademarks of CompuPro. CP/M is a registered trademark of Digital Research Inc. Americare is a trademark of Xerox Corporation.
System 816 front panel design shown is available from Full Service CompuPro System Centers only.

The Essential Computer™

SYSTEM 8/16
CompuPro.
CompuPro.
Hard Disk System

Circle number 1 on the reader service card.